Multi-Sensor Data Fusion and Modelling in Mobile Devices for Enhanced user Experience

By

Asmaa Hamoud Al-Marhoubi

A doctoral thesis

Submitted in partial fulfilment of the requirements

for the award of

Doctor of Philosophy

Department of Computer Science

Loughborough University

October 2016

Supervisors:

Professor Eran A Edirisinghe

Dr. Helmut Bez

Abstract

Recent advances in mobile technology have in particular being driven by user demand. Mobile handheld devices are now an important part of day-to-day life of every individual and thus the user experience provided by such a device is of significant importance.

Automatic Backlight Adjustment approaches are being used at present to enhance user experience when image/video content is being viewed under extreme illumination conditions. However such approaches suffer from content in some regions of the frame being completely destroyed due to the global adjustment of brightness that is made to vary dependent on illumination surrounding the mobile device display. A solution to this has been recently being proposed by a commercially patented technology, Iridix™. However this technology suffers from the shortcomings of the present generation Ambient Light Sensors (ALS) used for sensing illumination surrounding a screen, namely latency of operation and narrow angle sensitivity.

In this research we make use of the data gathered via various sensors available within a mobile device to accurately predict illumination within the 3D space surrounding a mobile device, even when a user is freely using the device. The data gathered from two motion based sensors, namely the accelerometer and gyroscope are used to model and predict the usage of the mobile device. This information is then used in identifying the exact motion of the mobile device within a 3D space that is lit up by a complex configuration of multiple light sources in order to model usage dependent illumination changes that a user experience around the mobile device. This way the more accurate values of illumination are determined that are used to enhance the performance of the Iridix™ technology, which in turn is able to significantly enhance the user experience when watching image/video content on a device screen.

The thesis investigates the use of various mathematical models and machine learning algorithms to serve the above purpose.

Asmaa, October 2016

Acknowledgment

This thesis represents the work of a journey of three and half years in the Computer Science Department at University of Loughborough. This acknowledgment is a special thanks to all who supported me in this journey.

First and foremost, I would like to thank my supervisor Prof. Eran A. Edirisinghe for his guidance, support, and encouragement through my entire Ph.D. journey. I would like to also thank my second supervisor Dr. Helmut Bez who has retired a year ago. He was kind and supportive. I would like to thank all the staff and colleagues at the Computer Science Department at the University of Loughborough. A special thanks goes to the company we had collaboration with for their support at the early stage of my research.

Secondly my thanks go to my friends Warda, Maryam, Seema, Eman and the rest of the Omani team. They are my sisters and they were supportive and stood by my side all along the journey. My thanks also go to my friend Orange (Zhao) for her help and support. Also Bob (Quang) for his kindness and help.

Where will I be without my family? My husband and son's caring, love and support made the Ph.D. journey possible. They both were by my side in good and difficult times. A huge thanks goes to my Father and Mother for their full support and encouragement. Their heart and soul were by my side even though we were miles away. They kept me in their prayers and provided me with endless love. I would also like to thank my in-laws for all the support they have given me during the journey.

Finally, I would like to give my gratitude to my country Sultanate of Oman for believing in me and giving me the opportunity to take my Ph.D.

Asmaa Al-Marhoubi October 2016

Table of Content

List of Figures

List of Tables

List of Abbreviations

Acronyms	Abbreviation
ALS	Ambient Light Sensor
MLP	Multilayer Perceptron
MA	Moving Average Filter
2D	2 Dimensional
3D	3 Dimensional
SSE	Sum of Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
RAE	Relative Absolute Error
K-NN	K-Nearest Neighbour
RF	Random Forest
USB	Universal Serial Bus
GPS	Global Positioning System
ABC	Automatic Brightness Control

Scholarly Contribution

The research presented in this thesis has resulted in publishing two conference papers. The details are as follow:

Marhoubi et al. (2015). The Application of Machine Learning in Multi Sensor Data Fusion for Activity Recognition in Mobile Device Space. *Image Sensing Technologies: Materials, Devices, Systems, and Applications. Baltimore: SPIE*, 1–9.

Marhoubi et al. (2015). Illumination Modelling of a Mobile Device Environment for Effective use in Driving Mobile Apps. *Image Sensing Technologies: Materials, Devices, Systems, and Applications. Baltimore: SPIE*, 1–11.

Chapter 1 : An Overview

1.1 Introduction

Advances of modern technology are driven by the users' demand for better quality products and services. Advances of mobile device display technology and image/video viewing applications are driven by user demand for high quality, precision and quality of experience. Not only consumers will want the images/video to be of the best quality, they would also demand that such excellent quality images and their viewability, dynamically adapt to the changes of the environment and user's behaviour/intentions.

For example, if the user is walking with a mobile phone in hand watching a movie or of some content of similar nature and is moving from indoors to outdoors where the illumination level suddenly increases, the user would demand that the screen adjusts its illumination to compensate for the extra light, thereby attempting to maintain user experience. A similar demand will be there if the user walks into a dark environment while watching a video. The clarity of content/detail, backlight level etc.; will play a key role in defining user experience. Adjusting the image by making it globally bright or globally dark is what is at present being attempted making use of the illumination level sensed by the Ambient Light Sensor (ALS) that is in-built in most modern mobile handheld devices. However this approach has a major drawback in that when a global increase in brightness is applied, already bright parts of the image are in the risk of being completely over exposed/*washed-off* loosing detail and clarity. At an opposite end if the brightness is decreased globally, already dark regions of the image could lose detail.

This unfortunately does not improve user experience but rather severely affects it. The impact to the user of such an attempt could vary depending on the user's intention of viewing and usage of the phone. For example if the user is watching a movie, seated, with full attention being given to enjoying the content, such

artefacts can severely impact user experience. However if the user is watching the movie while walking, sufficient attention may not be anyway given to the details of the content being watched and thus lack of details would not have a severe impact on user experience.

A commercial technology termed as Iridix™ was launched to resolve the above matter Criminisi et al. (2012). The idea was not to globally adjust illumination, but to adjust illumination locally, dependent on image content and details. This technology resolved the over or under exposure problems that results from ALS based global backlight/brightness adjustments. However its performance was affected by the current state of technology in ALS designs that suffer from two fundamental problems, narrow angle operation and latency of operation. Narrow angle operation of the ALS results in a sudden cut-off/reduction of the sensitivity of the ALS that affected Iridix™, as the illumination sensing was not sufficiently accurate. Further the latency of operation resulted in delayed changes being made to the picture based on the Iridix™ algorithm's performance. Both resulted in the non-optimal performance of the technology that would otherwise be promising. The delay will depend on the usage of the phone (e.g. if the user walking fast watching a video) and the distribution of light sources in an environment (e.g. which light source is where) as they could collectively contribute. If one could predict the usage of the phone (i.e. what is the user doing with the phone), it can be used to determine the environment and the mood in which it is being viewed.

To determine the environment in detail, one could attempt to model the illumination variation within the space in which the mobile device is being used so that a predicted illumination value could be used as a good starting point for the ALS reading, prior to actually reading it with a delay, due to the abovementioned shortcomings of existing light sensors. As the mood or usage of the phone will determine the level of attention that the user will put into noticing details of the scene being viewed, if the usage can be predicted that knowledge can be combined with the illumination modelling capability to provide the best user experience. Fortunately due to the availability of a large number of sensors

on a mobile device the sensor data can be fused to understand the intent/behaviour of a user and hence to feed into making Iridix perform optimally.

Modern mobile handheld devices have different number and different types of sensors that enables them to perform a variety of types of functions providing the best possible user experience. These sensors are categorized into three main categories; motion sensors, environmental sensors and position sensors (Android, 2014). One of the environmental sensors is the Ambient Light Sensor, which is considered to be one of the important sensors on portable devices in general. They are mainly used to adjust the screen brightness based on the surrounding environment. However it was explained above that simple approaches to screen brightness control have shortcomings that need overcoming. The demand for having a good quality of Brightness Control on handheld devices is increasing as time moves forward.

Some of the literature in the area of Automatic Brightness Control (ABC) has focused on controlling the brightness of the screen by using different methods. Bai (2012) proposed using a fuzzy logic controller to adjust the backlighting based on the environment lighting. Ma et al. (2012) proposed to make use of the front camera and a face detection algorithm to capture the users face along with the background and adjust the backlighting accordingly. Mailloux (2013) states that when the device wakes up after a sleep mode, it takes the first five samples every 400ms then after that it takes one sample after every 1.2 seconds. Whenever a new sample is received a new median will be calculated and the median sample values will be compared to the previous reading and the backlight will be adjusted accordingly, if it needs to change or remain as it is.

Given the above observations of the current state of art, in this thesis we combine two different capabilities of data modelling to optimize the usage of a current commercial technology Iridix™ that promises to resolve the shortcomings of the state-of-the-art ABC algorithms.

1.2 Research Motivation

In order to overcome the problems behind the effective and practical use of Iridix™, that promises to solve the shortcomings of the current ABC techniques, two capabilities should be developed.

1. The capability to determine the usage, behaviour and/or intent of the user of a mobile handheld device.
2. The capability to model illumination within the 3D environment of a mobile device.

These capabilities will resolve the two shortcomings of the current ALS devices, i.e. latency of operation and narrow angle of operation, thus resolving the challenges faced by Iridix™.

The presence of a multitude of sensors that can provide data to identify the phone usage and the illumination distribution in the environment in which a mobile handheld device is being used and the availability of advanced machine learning algorithms that can be trained to recognize such behaviour and variations provides the motivation behind the research to be conducted in this thesis.

1.3 Aim and Objectives

Aim: To enhance the quality of experience of a user of a handheld mobile device based on modelling phone usage, user intent and environmental conditions.

The specific objectives of the research are listed below:

- Conduct a comprehensive review of literature and commercial systems to determine the state-of-art in optimizing user experience when viewing image/video content.

- Study the design, operational details and performance limitations of various sensors integrated within the designs of the current generation mobile devices.
- Determine the usage of a mobile device by fusing together data captured from multiple motion sensors and learning behaviour through the use of advanced machine learning algorithms.
- Model the ambient illumination of an environment in which a mobile device is present by making use of mathematical and advanced machine learning based approaches.
- Make use of the motion sensor based behaviour/usage prediction capability to optimse the illumination modelling capability within rapidly changing environments and complex device usage applications.

1.4 Contribution of Research

The research presented within the context of this thesis has resulted in a number of original contributions. These can be listed as follows:

1.4.1 Sensor Fusion based Activity Recognition

In chapter 4, a method to predict the behaviour associated with a mobile device is presented that fuses together the raw component readings obtained from the Accelerometer and Gyroscope of the mobile device. The recognition of seven different activities that are frequently performed by most users of a mobile device is investigated. An in-depth analysis of using different machine learning algorithms for the above activity recognition that includes both single and ensemble learning approaches are investigated and presented. The results of this work was presented in an international conference Marhoubi et al. (2015) .

1.4.2 Mathematical Model based Illumination modelling in the Environment of a Simple Lighting Configuration

In chapter 5, the suitability of various mathematical models for the modelling and prediction of illumination variation within a 2D plane, in an environment that is lit up by one and two light sources are investigated. It has been shown that the mathematical models that are capable of accurate predictions are computationally costly and the simple mathematical models perform less

accurately. Out of the mathematical models tested the Sum-of-Sins demonstrated the best performance-cost balance and was recommended as a model that can be implemented in a computationally constrained environment. This work has been presented at an international conference Marhoubi et al. (2015).

1.4.3 Illumination Modelling in an Environment of Multiple Light Sources

In chapter 6, an investigation into the modelling of illumination in environments lit up by complex configurations of multiple light sources is carried out. It is shown that with increased complexity of light source configurations, the ability of simple, low computational cost, mathematical models to model illumination diminish. An initial investigation of possible use of machine learning algorithms and more complex mathematical functions to model illuminations is carried out that leads to recommendations being made to the suitability of each approach. The illumination modelling was carried out in a 2D, horizontal plane.

1.4.4 3D Illumination Modelling in an Environment of a Mobile Handheld Device

In Chapter 7, a method that integrates the motion sensor based behaviour/usage modelling approaches and ALS based illumination modelling approaches developed in the earlier parts of the investigations, were integrated to propose a novel approach to illumination modelling within a 3D space of an environment lit up by a complex, random configuration of multiple light sources. The suitability of using different machine learning algorithms for this purpose is investigated in detail. This work will be submitted as a journal paper to the IS&T International Journal of Optical Engineering.

1.4.5 Practical Solution to Limitations of ALSs

Considering all research contributions listed above which are outcomes of the fundamental and applied research conducted in this thesis, it is now possible to provide a complete practical solution for overcoming the limitations of present generation ALS, namely overcoming the latency of operation and narrow angled sensitivity. This research is at present being considered to be used in association

with Iridix™ technology for providing a more environmentally and usage adaptive, pleasing user experience on mobile handheld devices.

1.5 An Overview

For clarity of presentation, the research conducted within the scope of this thesis is organized into eight chapters, as summarized below.

Chapter 1: An Overview

This chapter introduces the reader to the research problem, state-of-art of research in the area, an overview to the solution being proposed and the key contributions of the thesis.

Chapter 2: Literature Review

This chapter presents related literature on mobile sensors. The first part presents research work that has focused on enhancing the brightness of the screen of handheld devices using a number of different techniques. The second part of the chapter presents research that has made use of multiple sensors that are available on handheld devices for activity recognition. An additional section provides a comparison between different classification algorithms.

Chapter 3: Background

This chapter presents the theoretical background of six different mathematical models proposed for modelling the ALS readings. It also provides an introduction to the different mobile phone sensors used in this project and the machine learning algorithms used for modelling and behaviour/usage prediction.

Chapter 4: Sensor Fusion for Activity Recognition

This chapter presents the use of multiple sensors to facilitate identifying the different activities performed by the user. An investigation into the use of different Machine Learning algorithms for the purpose is carried out.

Chapter 5: Modelling of Illumination

This chapter investigates the use six different mathematical models for the prediction of illumination in the environment of one and two light sources based on the illumination values sensed by the ALS.

Chapter 6: Illumination Modelling in an Environment of Multiple Light Sources

This chapter investigates the use of more complicated mathematical models and machine learning algorithms to predict the illumination variation within a horizontal plane of a more complex arrangement of light sources.

Chapter 7: Multi Dimensional Illumination Modelling & Prediction using ALS, Gyroscope and Accelerometer

This chapter investigates the use of motion sensor based data gathering and ALS readings for the modelling of illumination within a 3D space, under complex lighting configurations and mobile phone usage patterns.

Chapter 8: Conclusion

This chapter concludes the research conducted within the scope of this thesis and provides an insight into the future work that can be carried out.

Chapter 2 : Literature Review

2.1 Introduction

This chapter presents a review of existing literature in the areas of main research foci of this thesis, namely the activity recognition of a mobile handheld device and the illumination prediction in the environment of a mobile device. In the proposed research both research problems are approached based on the fusion of data captured by the numerous sensors available within a mobile device. In Chapter-1 it was discussed that the successful solutions to the above problems will help in our ability to improve the usability of a mobile device by adopting the present generation mobile Apps respond appropriately to user behaviour, intent and the environment in which the phones are being used. This will significantly help enhance the user experience.

For clarity of presentation this chapter is divided into several sections. Section 2.2 provides a detailed literature review of existing research in user activity recognition based on using different sensors and using different methods to identify the said activities. As prediction of illumination in the environment of a mobile device is not an area investigated in literature to-date, section 2.3 provides details of the research closest to this area, i.e., on different methods used for automatically adjusting the backlight of a mobile phone based on the illumination sensed from the environment in which the phone is being used. Finally section 2.4 concludes the review of literature with an insight to identifying the gaps in existing research and indicating how these research gaps are investigated within the context of the research presented in the contributory chapters of this thesis, i.e. chapter 4-7.

2.2 Activity Recognitions using Mobile Handheld Device Sensors

In chapter-1 it was mentioned that present generation handheld devices consists of a significant number of inbuilt sensors that are aimed to be used to enhance the experience of a mobile handheld device user. In most instances theses

sensors are used either individually or in pairs for basic activity recognition that can subsequently be used to enhance the performance of a mobile App.

The following is a review of literature on activity recognition in the domain of mobile devices. Many different approaches have been used based on the sensors used, methods adopted for the processing of raw-sensor data, types of features extracted and used, recognition techniques adopted, etc.

In majority of the cases it is only the accelerometer that is used Awan et al. (2012), Kwapisz et al. (2011) and Coskun et al. (2015). In some of the cases that only use the accelerometer, only one directional component of the accelerometer (depending on the style of activity) has been used Thammasat (2013). However, Ustev et al. (2013) have used the accelerometer, gyroscope and magnetometer but had the phone kept in the pocket while performing the activities and each activity was performed independently.

Coskun et el. (2015) Investigated the possibility of identifying the location of the phone with respect to its bearer, using only the Accelerometer sensor. Data was collected from fifteen users, each of whom were requested to carry out certain described activities, while placing three different phones in three different locations namely; holding in the hand, placed in a carried bag and in the pocket. All users performed six activities namely, sitting, standing, walking, running, ascending and descending the stairs and travelling in a bus. The learning algorithm Random Forest was used for the classification purpose obtaining an average accuracy of 85%. Yang et el. 2014 used the only the Accelerometer sensor of a mobile handheld device to identify the activities of the user, including lying down, sitting, standing, walking, running and ascending/descending along a flight of stairs. The authors claim that a classification model has been used for activity recognition but did not specify which particular algorithms were used. Accuracy levels of 87.1%, 75% and 90.3% were reported for walking, sitting and standing respectively.

Anjum et al. (2013) built a mobile application that estimates calories burnt by a user based on different track-able user activities performed by the user. The accelerometer sensor values were used for the purpose. The physical activities recognized include walking, running, ascending and descending stairs, cycling, driving and finally being inactive. They have compared between using four different Machine Learning algorithms namely Naïve Bayes, K-Nearest Neighbour, Decision Tree and Support Vector Machine. It was proven that the Decision Tree C4.5 classifier gave the best accuracy value of 95%. (Kwapisz et al. (2010) on the other hand, proposed the use of the mobile device's accelerometer to identify the user's behaviour, including, sitting, standing, walking, jogging, ascending stairs and descending stairs. The phone was kept in the users pocket while the experiments were carried out. The authors compared the use of three different classifiers, including the J48 classifier, Logistic Regression and Multilayer Perceptron with the Multilayer Perceptron providing the best results.

In majority of the cases it is only the accelerometer that is used Awan et al. (2012), Kwapisz et al. (2011) and Coskun et al. (2015). In some of the cases that only use the accelerometer, only one directional component of the accelerometer (depending on the style of activity) has been used Thammasat (2013). However, Ustev et al. (2013) have used the accelerometer, gyroscope and magnetometer but had the phone kept in the pocket while performing the activities and each activity was performed independently.

Ustev et al. (2013) investigated the recognition of activity performed by the user of a mobile device, using three sensors namely the accelerometer, magnetic field and gyroscope sensors. The focus was on the recognition of five activities namely, sitting, standing, walking, running and riding a bike. Twenty users performed the activities by using the phone in different orientations. The phone captured and fused the data from accelerometer, magnetic field and gyroscope sensors. They extracted features from the accelerometer and used it along with K-Nearest Neighbour Algorithm for analysis. The aim of using multiple sensors is to overcome the orientation problem of the phone during performing the activity.

Kaghyan et al. (2012) implemented and used K-Nearest Neighbour Algorithm to identify between six different activities, performed by the user. The activities are; siting, standing, ascending and descending stairs, walking and running. Only data from the accelerometer were used in this classification. The accuracy levels obtained for simple activities such as standing and sitting were high but those reported for all other activities were low.

Awan et al. (2012) proposed enabled identifying user activities through data gathered from wireless sensor technologies that were connected to a backend server. The use of three machine learning algorithms, namely, Naïve Bayesian, J48 and Logistic Regression classifiers were investigated for modelling. The data was collected using the accelerometer only and activities such as sitting, standing, walking and jogging were investigated. The research proved that the Naïve Bayesian algorithm performed the best. Y. Ma et al. (2012) proposed a framework to identify the mood of a person in daily life using a variety of sensors. As a part of the assessment, the research attempted to identify user's activities including sitting, standing, walking and running. Only the accelerometer was used for data collection.

In the cases machine learning algorithms have been used, only few candidate algorithms have been tested and the best amongst the tested algorithms have been recommended. Patil (2013) and Sharma (2011). Experiments have revealed that the suitability of the algorithms depends on the data gathered Aydemir et al. (2013).

Both statistical and machine learning based algorithms have been used for activity recognition. Thammasat (2013) conducted a statistical analysis of data gathered from the mobile devices accelerometer to recognize three different activities, namely walking, running and jogging. The approach proposed was not able to distinguish between running and jogging but was able to separate and identify walking from the other two activities. However, the machine learning based approaches have proven to provide better results.

Most existing algorithms are based on capturing the sensor data, extracting specific features from the sensor data and then subsequently using this data in training a machine learning algorithms. Extracting features from the raw sensor data can be time consuming and hence may not be applicable to be used in computationally constrained application area.

2.3 Automatic Phone Backlight Luminance Adjustment Algorithms

Adjusting the right level of backlight luminance of a display for the comfort of the user's eye and for prolonging the battery life of a mobile handheld device is a Research & Development challenge that researchers have investigated recently. Typically, the backlight level is adjusted based on the reading of the ALS. Unfortunately, modern day ALSs requires few seconds to respond to any new environment, i.e. if the illumination changes from darker to brighter or vice versa. In literature a number of research attempts have been made in proposing different solutions to automatically adjusting the backlight brightness. This work has been summarized and reviewed below.

A number of different approaches to backlight adjustment have been proposed in literature and used in practice. The most common solution is to use the ambient light sensor to sense the illumination of the environment and accordingly changing the brightness of the screen by an appropriate (i.e., proportional) adjustment of backlight.

Na et al. (2014) developed a dynamic backlight luminance adjustment approach based on a study carried out for the determination of the eye comfort of a user. Their target was to identify the best luminance level for the initial and continuous viewing in a dark environment. Five levels of luminance were defined for the screen to change to. The comfort/discomfort of a user was investigated and measured based on the frequency of blinking, users' preference, and discomfort by luminance intensity and change of facial expression etc. The

authors concluded that the best luminance for initial viewing is 10 cd/m^2 and for constant viewing is 40 cd/m^2. Another method was proposed by Ma et al. (2012) for enhancing the illumination. They proposed an automatic brightness control algorithm for low illumination environments. The front facing camera of the phone was used to take an image of the face of the mobile device user. Subsequently a face detection technique was used to calculate the contrast ratio between the user's face and the background. Authors used this ration to identify the best brightness level of the screen. Na et al. (2014) method depends on the blinking of the user, which can be effected by the condition of the eye (example dry eye will result in more blinking). On the other hand, Ma et al. (2012) method has the drawback of high battery consumption due to the useage of the camera.

Bai and Cheng (2012) proposed the use of a fuzzy logic controller to automatically adjust the backlight brightness in a mobile device based on the environment where the phone is placed. The focus was on increasing the comfort of the eyes and reducing the power consumption of the handheld device. The detection of the illumination level of the environment is done every second and the screen brightness was adjusted accordingly if any change occurs. Bai and Cheng (2012) covered the drawbacks of Ma et al. (2012) and Na et al. (2014). However, Bai and Cheng (2012) method does not solve the two drawbacks of ALS that are latency of operation and narrow angle.

2.4 Summary & Conclusion

The literature review concludes that it is vital to consider all data from a sufficient number of available sensors that will help identify the required user activities. The use of more types of sensor data will help improve the discrimination power of the sensor data. Further due to the additional computational cost of extracting middle level features from the sensor data prior to conducting training, may un-necessarily increase computational cost that will result in such secondary data based learning approaches being infeasible for practical implementation in a resource constrained environment, for e.g. use within a mobile phone. Further as the prediction accuracy of different machine learning algorithms are data dependent, it will be recommended to carry out an

in-depth and rigorous study of the use of a wider group of learning algorithms. These conclusions have led to the main aim of the research presented in this thesis, i.e. to develop an efficient, accurate, multi-sensor based machine learning approach for activity recognition of mobile devices based on a rigorous study of suitability of different algorithms for different type of activity recognition. The proposed approaches therefore use all data that is captured by both the encoder and decoder, in activity recognition.

The literature review in the area of illumination prediction in the environment of a mobile device showed that only an insignificant amount of research exists in this area. Few researches have been conducted so far to model the illumination variation behaviour/pattern of any lit up area, in three dimensions. Hence the available systems cannot accurately and sufficiently quickly adjust the backlight illumination of a screen when the lighting within the viewing area suddenly changes. The typical narrow angle operation of the ALS sensors and their latencies of operation result in delayed adjustment of screen backlight illumination, resulting in a sub-optimal user experience, when using a mobile device in particular in a fast changing environment, where the illumination levels can significantly vary. Although few of the past techniques suggested the use of face recognition techniques via the use of a mobile device camera to measure the illumination changes between the face (i.e. the foreground) and the background areas, such systems are not recommended to be used within mobile devices due to their high cost on the mobile device battery longevity. Given the above observations in this thesis, rigorous and comprehensive background research is conducted into modelling illumination within a 3D phase using a large number of different learning algorithms. Recommendations are made as to what algorithms can perform best in practice.

Having concluded the review of literature, in Chpater-3, the fundamental concepts, systems and theories upon which the research in the contributory chapters of this thesis is built up, is presented.

Chapter 3 : Background

3.1 Introduction

This chapter presents the important theories, concepts, definitions and systems that are needed to understand the content of the chapters that follow. It presents in detail the six different mathematical models that were designed and implemented for the purpose of modelling illumination distribution/variations and the various evaluation metrics used in the comparison of their performance.

The chapter also presents details of the various sensors that are available in a modern mobile handheld device, in particular those that are used in the proposed research for data collection. Further details of the machine learning algorithms used in Chapter 4-7 in modelling the sensor fusion and the illumination variations are also discussed. Finally the chapter presents some details of the special software developed as a part of this research (Euler software) to capture data from all mobile handheld device sensors.

3.2 Mathematical Models

A mathematical model is said to be a description of, for example, a chemical, physical or biological process or state. Such a model provides the ability of modelling the variations and trends of the changes of the physical feature it represents, using a mathematical formula, allowing one to be able to predict the value of the said feature, at a previously non-considered state or in the future.

For example let's consider a person walking down a corridor that is lit up by a single light source or an array of multiple light sources placed in an ordered or non-ordered manner, placed on the ceiling. While the user walks down the corridor he/she will experience changes of illumination. The changes of illumination and the rate of change will not only depend on the light source intensities and their pattern of distribution but also on the pattern of movement of the object or person experiencing the changes. One would assume that if only one light source with fixed intensity is present and the illumination changes experienced directly under the source will follow a normal distribution as the illumination intensity from a light source in known to vary inversely

proportional to the square of the distance from a point light source. However fluctuations of the light sources illumination, other light sources present and reflections etc., may cause unexpected changes to the otherwise smooth variation.

In the research conducted in chapters 5-7, modelling the illumination variation of an environment lit up by multiple light sources becomes an essential requirement. Thus six different mathematical models that are likely to be able to accurately model the illumination variation are presented in this section. They include, the Sine, Fourier series, Polynomial, Sum of Sine, Smoothing Spline and Gaussian functions. It is noted that each model will result in a different level of goodness of fit (see Chapter-5) for the purpose of illumination modelling and that each model will have a different level of mathematical complexity. Both the goodness of fit for the purpose and the mathematical complexity will influence a given model's practical suitability for their use in practice.

The functions can be defined as follows:

3.2.1 Sine Function

Sine is defined as a trigonometric function of an angle. "The sine of an angle is defined in the context of a right angled triangle: for the specified angle, it is the ratio of the length of the side that is opposite that angle to (divided by) the length of the longest side of the triangle (i.e. the hypotenuse)."(Bourbaki et al. (1994). Alternatively a Sine wave "is a mathematical curve that describes a smooth repetitive oscillation" (Wiki ,2015).

$$f\ (x) = a \sin\ (bx + c) \qquad (3.1)$$

Where 'a' is defined as the amplitude or the peak of the wave, 'b' defines the length of one full oscillation along the x-axis and 'c' defines the phase shift, horizontally.

3.2.2 Gaussian Function

The Gaussian function graph is represented in a bell shaped curve. Comparing it with the actual graph of the illumination variation as measured in the proposed experiments; there is a possibility of the mathematical model well fitting the variations of illumination. This is why the Gaussian distribution was considered as one of the candidate functions for fitting the illumination variations.

A Gaussian is represented by the following formula:

$$f(x) = a \exp\left(-\frac{(x-b)^2}{c^2}\right) \qquad (3.2)$$

The parameter 'a' represents the peak of the curve, while the parameter 'b' represents the position of the center of the peak. Parameter 'c' is responsible for controlling the width of the bell (Guo, 2011) shape of the distribution.

3.2.3 Fourier series

Fourier series is a periodic function f (x) with an infinite series of sum of sine and cosine (Pearce, 1944) terms and can be represented by the following equation. The 'a' and 'b' values define the impact of different harmonics, defined by 'n'.

$$f(x) = a_0 + \sum_{n=1}^{\infty}(a_n \cos nx + b_n \sin nx) \qquad (3.3)$$

3.2.4 Polynomial Functions

Polynomials are used for the approximation of physical concepts as they can be differentiated, evaluated and integrated easily by using basic arithmetic operations. In a polynomial function, the 'degree' refers to "the highest exponent to which the dependant variable is raised."(Wiki, 2014). An n-th order polynomial can be defined as follows:

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0 \qquad (3.4)$$

3.2.5 Sum of Sine

It is a model that fits well with periodic functions and is defined as:

$$y = \sum_{i=1}^{n} a_i \sin(b_i x + c_i) \qquad (3.5)$$

The amplitude is defined by 'a' and the frequency is represented by 'b' and the phase constant for each sine wave is defined by 'c'. Further 'n' represents the number of terms in the series.

3.2.6 Smoothing Spline

Smoothing Spline is defined as a method of smoothing which uses a spline function. A spline is a numeric function that is defined by polynomial functions. It has a high degree of smoothness in the points where the polynomial functions connect. It can be very complex but at the same time a smooth and flexible function. Splines are in particular useful in modelling illogical functions (Boore, 1978).

Implementation of smoothing spline on mobile hardware components is complex due to its nature of functionality. Assuming a set of values is presented in a graph. When smoothing spline function is implemented, it will generate a linear function for each link between two presented values in the graph (Boore, 1978). This will result in having multiple linear functions, each representing the link between two different values. In the time of implementation on hardware components, a large number of liner functions will be considered which will result in having long processing time.

3.3. Model Evaluation Metrics

This section summarize the different metrics used in chapters 4-7 to help evaluate the effectiveness of the various mathematical models, introduced in section 3.2 for the purpose of modelling illumination variations.

3.3.1 Sum of Squared Errors (SSE)

SSE measures the total deviation of the actual values from the predicted values and can be defined as follows:

$$SSE = \sum_{i=1}^{n}(x_i - \bar{x})^2 \qquad (3.6)$$

Where x_i represents the predicted values and $\bar{x}$ is the mean of the actual values.

3.3.2 R-Square (Coefficient of Determination)

The R-Square value represents the square of the correlation between the original set of values and the predicted values. It is given a value between 0 and 1. The closer it is to 1 it indicates that it is more accurate (Mathworks, 2014).

$$R^2 = 1 - \frac{SSE}{SST} \qquad (3.7)$$

Where SSE is defined as in section 3.3.1 and SST is the Sum of Squared Total that is defined as follows:

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2 \qquad (3.8)$$

Where $\bar{y}$ is the mean of the actual values and y_i is the predicted value.

3.3.3 Adjusted R-Square

The Adjusted R-Square value takes the statistics from the R-Square value and adjust it according to the residual degrees of freedom, which is defined as the number of response values, 'n', minus the number of fitted coefficients, 'm', estimated from the response values, v = n-m. V indicates the number of independent pieces of information involving the 'n' data points that are required to calculate the sum of squares (Mathworks, 2014).

$$R^2_{adj} = 1 - \frac{SSE(n-1)}{SST(v)} \qquad (3.9)$$

Where, SSE is defined as in section 3.3.1 and SST is defined as in section 3.3.2, above.

3.3.4 Root Mean Squared Error (RMSE)

RMSE is the most common metric used to measure errors (Witten I. at el. 2011). It shows how close are the actual values to the one predicted. RMSE is said to be an absolute measure to fit. If prediction is the main purpose of the model, RMSE is useful in measuring how accurate the model predicts. The closer the value is to 0, it means that the fit is better (MathWorks, 2014).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{a}_i - a_i)^2} \quad (3.10)$$

Where $\hat{a}_i$ is the predicted value and a_i is the actual value at time i.

3.3.5 Correlation Coefficient

The Correlation Coefficient measures the linear relationship between the input variables and the targeted variables. Its value is between -1 and 1. A positive value means that the two variables move in the same direction with respect to their means. On the other hand, a negative value means that they are moving in opposite directions with respect to their means. A value close to 0 means that the two variables have little linear dependency.

3.3.6 Mean Absolute Error (MAE)

The MAE is the sum or individual absolute errors, normalized by the number of samples. Individual error is defined by the difference of ground truth and predicted value for a sample.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{a}_i - a_i| \quad (3.11)$$

Where $\hat{a}_i$ is the predicted value and a_i is the actual value at time i.

3.3.7 Relative Absolute Error (RAE)

The RAE is the total of the absolute errors, normalized.

$$RAE = \frac{\sum_{i=1}^{n}|\hat{a}_i - a_i|}{\sum_{i=1}^{n}|\bar{a}_i - a_i|} \qquad (3.12)$$

Where $\hat{a}_i$ is the predicted value and a_i is the actual value at time i.

3.4 Moving Average Filtering

This filter is known for its simplicity and is used with time series data in order to smoothen the minor fluctuations in the graphs. It is ideal for reducing random noise. It is defined as a calculation used to analyse different data points by taking the whole data set and dividing it into small, fixed size subsets. For the first subset, an average will be taken then its values will be modified. Then it will move to the next subset and will exclude the first number in the series and it will add following number from the next subset and get the average (Smith, 1991). In the proposed research the Moving Average filter has been used to reduce noise in the raw sensor data (See Chapter 4).

3.5 Mobile Sensors

Before the existence of smartphones, the interaction with sensors on a daily basis was somehow limited for most consumers of mobile devices. Sensors were usually designed to serve a single specific purpose such as television remote control. However, sensors were not available as much as they currently are, to aid in improving user experience of mobile devices. However at present, most handheld mobile devices come equipped with multiple sensors that enhance the use of the device.

The vast majority of handheld devices come with a large number of sensors. Ambient light, gyroscope, accelerometer, compass and proximity sensors are considered as the key in-built sensors within a smartphone. These sensors are mainly categorized into three main groups namely, Motion sensors, Environmental sensors and Position sensors. Motion sensors are used to measure rotational and acceleration forces along three axes, Environmental

sensors measures environmental parameters and Position sensors measures the physical position of the handheld device (Wiki, 2014).

For handheld devices, the sensors have been miniaturized physically to match with the size of the devices as the devices are becoming smaller and thinner, with time. Such sensors are called Microelectromechanical sensors (MEMS) and are usually placed on a silicon chip. The sensors can be categorized into two types: synthetic and raw or physical sensors (Nagpal, 2016). Raw sensors provide raw data that comes from actual physical sensors placed on the handheld device. Synthetic sensors on the other hand, can modify the data of the raw sensor to make it easier to use in addition of combining raw data of multiple sensors. Some of the sensors that provide raw data are light, proximity, magnetic field, accelerometer and gyroscope sensors. Examples of Synthetic sensors are orientation, linear acceleration, gravity and rotation vector (Greg Milette, 2012) sensors.

In general, sensors may output a binary value or a value that ranges between a minimum and maximum, with a set number of intermediate levels. Given this a sensor output can be represented by a binary value made out of either one or more bits.

3.5.1 The Mobile Coordinate System

The coordinate system on the phone uses six different degrees of freedom. This means that the device is free to move forward/backward, up/down and left/right. It works together with pitch (tilt forward and backwards), roll (tilt on sides) and yaw (turn left to right) as illustrated in the figure below (Greg Milette, 2012).

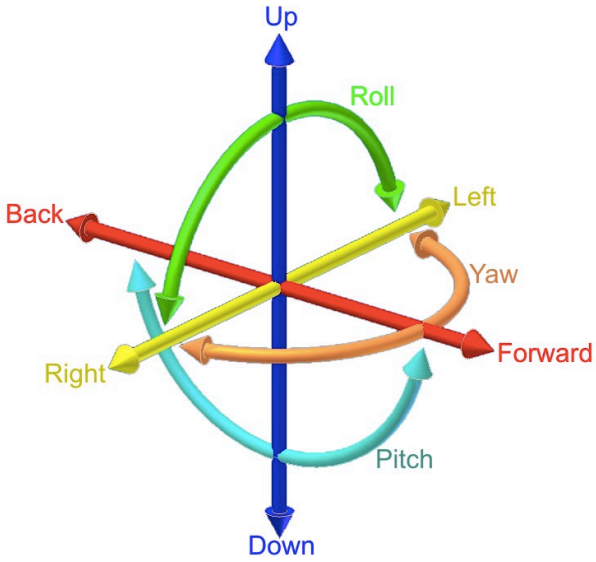


Figure 3.1 The six different degrees of freedom (Wiki, 2014)

The coordinate system for most of the sensors is defined based on the device screen as shown in the figure 3.1. The x-axis is horizontal to the screen of the phone with positive values on the right. The y-axis is vertical to the screen of the phone with positive values towards the top of the screen. The z-axis is a line from the centre of the screen towards outside having positive values (Greg Milette, 2012).

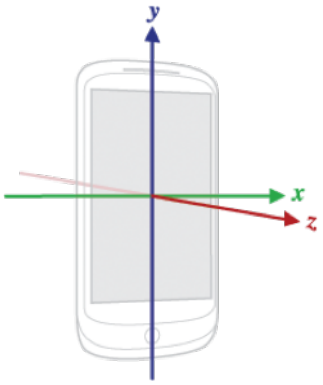


Figure 3.2 Coordinate system relative to a device (Android, 2014)

One important point to keep in mind is that even if the orientation of the device changes, the axes are not going to be swapped and will remain the same (Android, 2014).

3.5.2 The Sensors

A number of different sensors are available on mobile handheld devices (see Table 3.2) and the types and exact number of sensors present in a device may vary according to the make, model and price of the device. The sensors that are being considered to support the research conducted within this thesis includes the ALS, Gyroscope and the Accelerometer. Further details of these three sensors are as described in table 3.2.

Table 3.1 List of common sensors along with the parameters of each sensor (Android ,2014)

Sensor	Parameters
Accelerometer (including force of gravity)	X, Y & Z
Gravity (measures the force of gravity)	X, Y & Z
Gyroscope	X, Y & Z
Linear Acceleration (excluding force of gravity)	X, Y & Z
Magnetic Field (for creating a compass)	X, Y & Z
Rotation Vector (motion and rotation detection)	X, Y & Z

3.5.2.1 Ambient Light Sensor

Ambient light sensors are used for measuring the level of illumination of the environment in which a handheld device is placed. The readings sensed from the sensor are used to automatically to change the brightness of the screen, i.e. to make it dimmer or brighter. The ambient light sensor is usually placed at the front top of the screen. The window that covers the sensor is completely transmissive to allow light to pass through.

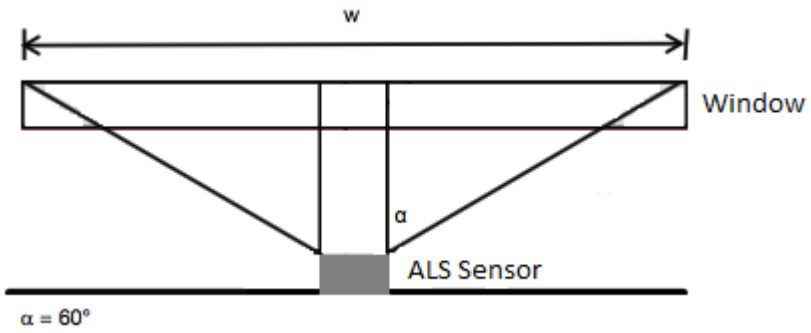


Figure 3.3 Window size measurement (Vishay, 2012)

To enable the sensor to give its best performance, the size of the window should be large enough so that enough amount of light can get into the sensor as illustrated in figure 3.3. When designing the window size, the only important dimensions to be considered are the distance from the top surface of the ALS to the outside part of the window. Along with that is the size of the window as it determines the detection zone (Vishay, 2012).

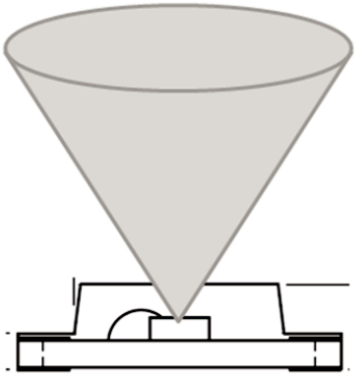


Figure 3.4 ALS sensing part is cone shaped (Vishay, 2012)

As presented in figure 3.4 the sensing part of the ALS is in a cone shape. Consequently, rotating the phone around the z-axis in a flat surface will not affect the illumination value being captured. However this is not the case if any rotation around the x and y-axes are to be accommodated.

The light sensor on handheld devices is used to measure the illumination in units of 'lux' (Nagpal, 2016). It uses a basic concept where it changes the status of the screen to become 'dim, if the surrounding environment has low illumination, but if the illumination is high, the brightness of the screen will be increased to make it comfortable for the eye to still see what is on the screen. This concept still does not work as accurate as it should (Display Mate, 2010) and will cause unnecessary use of battery power in environments that are brightly illuminated. However in low illumination environments the ALS will have a positive impact on battery usage (Sensor, 2009).

3.5.2.2 The Accelerometer Sensor

The Accelerometer is used to identify the accelerations along the three axes, x, y and z by calculating the force based on the phone movements. It is a low power

consumption physical sensor that generates raw data. On handheld devices, the functionality of the accelerometer depends on the movement of microscopic crystal plates (i.e. a seismic mass) over a microscopic crystal base, in x, y and z directions. When the phone moves due to force generated by the user, this results in the movement of the seismic mass, touching the microscopic crystal base. This action will result in a change in the capacitance, which will use electric circuits to convert the change into force. The values generated by the accelerometer are measured in (m/s^2). If the phone is kept still and in a horizontal plane, the acceleration along the z-axis will be $9.81 m/s^2$ which represents the acceleration due to gravity (Nagpal, 2016).

3.5.2.3 The Gyroscope Sensor

The Gyroscope is used to provide the rate of rotation of the handheld device along all three axes (Nagpal, 2016). In a handheld device, the gyroscope is made of two capacitor plates. "One capacitor plate is fixed to the frame and one is fixed to the moving mass (Greg Milette, 2012). This movement will result in change in the capacitance. Electrical signals will be used to convert the change of capacitance into rate of rotation. The force in the Gyroscope happens only when the handheld device is being rotated. Consequently, the Gyroscope measures the speed of rotation of the handheld device. If the device is not moving then all the three axis's x, y and z will report the value zero regardless of the position of the phone (Greg Milette, 2012). The Gyroscope is capable of providing calibrated and un-calibrated values. The values generated by the Gyroscope are measured in radians per second (rad/s) (Nagpal, 2016).

3.5.3 The Sensor Value Types

Sensors may generate three different types of values. These include raw, calibrated and fused values. Raw values are the readings generated directly by the sensor and transferred by the operating system directly to the application using the data, without any corrections done to it (Nagpal, 2016). The calibrated values of the sensors are subjected to a correction algorithm by the operating

system. An example of the reason of using a correction algorithm is to remove noise attached to the raw sensor values. The sensors that generate calibrated values such as the step detector usually rely on base sensors such as the Accelerometer. Certain sensors are capable of generating both raw and calibrated values, e.g. such as in the case of the Gyroscope (Nagpal, 2016). The third type of sensor values is called the fused values. It is a combination of values that are generated from one or more sensors. Fusing the values usually happened if one of the sensors is not very strong, while the other one is, combining them or fusing them will result in having a stronger result. An example of it is the linear acceleration sensor. It is a result of fusing the gyroscope and accelerometer values (Nagpal, 2016).

3.6 Machine Learning Techniques

Machine Learning (ML) is a process of learning useful information from a large amount of data. This learning process will lead to making intelligent decisions for the future based on past and upcoming data. Therefore, ML has the ability to develop methods or tools to discover un-seen patterns from data to solve a particular task or problem. By building a suitable model to fit with the already observed data predictions for the future can be made.

In general, supervised machine learning processes involve two phases; learning or training and testing or validation. A number of supervised ML algorithms exist with each having different features and approaches to training, modelling and prediction. However, they all do have the same starting part, which is training the data, and the same ending part, which is the creation of the predictive model. In general, the machine learning algorithms can be divided into two categories; the single learner algorithms and ensemble learner algorithms.

3.6.1 Single Learning Classifiers

A single learning classifier is a Machine Learning technique that takes the whole training data set, without dividing it and applies a single Machine Learning algorithm to it to enable building up the prediction model.

3.6.2 Ensemble Learning Classifiers

Ensemble Learning is an approach that uses different classification techniques to build up a model. The data will be divided into multiple sub-sets depending on the classifier used. Each set will go through a learning/ training process and a model will be generated. In other words, each set will generate a model based on a selected base classifier. The models will subsequently go through testing. [Note: In the proposed work ten-fold cross validation was used.] The best model will be chosen based on a voting/ averaging technique, depending on the ensemble learning algorithm type. Finally the prediction model will be generated (Witten I. at el. 2011).

3.6.3 Cross Validation

Cross validation divides the data received into multiple folds (default is 10, in WEKA). In the first round, data in one fold will be used for testing and the data in the rest of the folds will be used for training. In the next round, data in another fold will be used for testing and the rest of the data will be used for training. The process will be repeated until all folds go through testing and contribute towards training. Finally the average prediction accuracy will be calculated over the entire dataset. Cross validation minimizes the chance of bias of the test-training datasets and provides a fair result, independent of chance.

Research conducted on different data sets using different learning techniques have proven that 10 folds is the optimal number of folds to use for the best estimation of errors (Maloof, 2002). Therefore in all experiments conducted in this thesis 10 fold cross validation was used.

3.6.4 Classification vs. Regression

In literature, often the terms classification and regression are wrongly used to describe a machine learning procedure. “A classifier is an algorithm which has to be trained with labelled training examples to be able to distinguish new unlabelled examples between a fixed set of classes”(Aydemir at el., 2013). Classification is the method used to find a model that distinguishes data classes.

The created model is developed based on the training data set. Regression on the other hand, is used to predict values that are missing or that are not available (Han at el., 2012).

3.6.5 The Classifiers

As stated in section 3.6.1 and 3.6.2, there are two types of classifiers, namely, single and ensemble learning classifiers. In this section, the classifiers used in the research conducted in this thesis are summarized.

3.6.5.1 Bagging

Bagging is an ensemble learning classifier that increases the accuracy of the machine learning algorithms. Bagging (Breiman, 1996) also called as bootstrap aggregation, was developed by Leo Breiman to enhance the accuracy of single learner classification algorithms.

In Bagging, the original set of data n is divided into m subsets (bags). n' Represents the instances chosen randomly from the original set of data n. Some of the n instances can be duplicated in the m subset. The data n' is distributed randomly in each D_m subset (bag), where $n' < n$. There is a possibility that an instance is duplicated in the same subset. Each subset will train the data and generate a model. The result models will be combined together to give an output (Breiman, 1996). The parameters used for bagging are bag size percent is 100, number of iteration are 10 and seed is set to 1.

3.6.5.2 Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) or a Neural Network is a simple model of the brain functionality. The basic structure of a Neural Network is that it has an input layer, hidden layer/s and an output layer. The input represents the data that needs to be processed. The hidden layer/s is where the processing of the data happens. It contains a number of nodes and each node has a unique equation. The weights are different for each one of them. Once the result is produced, it is error checked, i.e., the actual data is compared to the predicted data. If the error is high, then the weights will be modified to enable a reduction of error and to achieve higher accuracy. More than one hidden layer can be

added but this will have an impact on the complexity algorithm. The parameters used for MLP are hidden layers set to 3, learning rate set to 0.3, momentum set to 0.2 and training time 500.

3.6.5.3 Random Forest

Random Forest is an ensemble learning method that can be used in both regression and classification. Assuming that each classifier is a decision tree classifier, the collection of them represents a forest. Each tree is generated using a random selection of attributes in each node to identify the splits. At the time of classification, each tree will vote and the class voted the most, will be returned (Han at el., 2012). The parameters used for Random Forest are number of trees set to 100 and seed is set to 1.

3.6.5.4 J48 Classifier

J48 is an algorithm that generates decision trees and is a slightly improved version of the C4.5 (Witten I. at el. 2011). The C4.5 classifier is an improved version of ID3 algorithm developed by Ross Quinlan (Quinlan, n.d.).

J48 generates an initial tree using a divide-and-conquer algorithm based on an 'n' set of cases. The divide-and-conquer algorithm works as follows (Panda & Patra, 2008):

- If 'n' is small in size or all the available cases belong to the same class 'n', the leaf of the tree will be labelled with the most frequent class.

- If that's not the case, then it will choose a single attribute that has two or more subsets. It will make the single attribute the root of the tree and the subsets as leafs. Each leaf will be divided into subsets until the subsets are pure and cannot be divided anymore.

The parameters used for J48 are min number of obj is set to 2, number of folds is set to 3 and seed set to 1.

3.6.5.5 K-Nearest Neighbours (K-NN)

KNN is a well-known classifier that is widely used in pattern recognition. It is a simple classifier that does not use a model but relies on memory instead. K-NN is considered to be complex if the data set is large due to need to search the distance for the whole data set (Kaghyan at el., 2012).

Assume two given classes (a, b) and an object 'c' for which one needs to determine the class to which it should belong. If k=3, Object c will find the three nearest neighbours in both classes, a and b. The class with more neighbours pointing closer to c will be the class that c will belong to. If the number of classes is even, then the value of k should be odd to avoid a tie. K represents the number of neighbours used (Kaghyan at el., 2012). The parameters used for KNN are nearest neighbour search algorithms set to linear NN search, distance function set to Euclidean distance and KNN set to 1.

3.6.5.6 Naïve Bayes

The Naïve Bayes classifier is a simple probabilistic algorithm that uses Bayes theorem and Naïve independence assumptions (Aydemir at el., 2013). The functionality of Naïve Bayes works by taking the data set and converting it into a frequency table. This table represents the number of times the term x (i.e., word) appears in a document y.

The second step is to find the probabilities and creating a likelihood table based on the frequency table. (See example in figure 3.3). An example of the probabilities in the likelihood table is the probability of being 'rainy' is 0.36. The last step is to calculate posterior probability for each class using the Naïve Bayes equation. The prediction will be the class with the highest posterior probability (Sunil Ray, 2015).

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Figure 3.5 An example to illustrate how Naïve Bayes algorithm works (Sunil Ray, 2015)

The parameters used for Naïve Bayes are debugging set to true and use kernel estimator set to true.

3.6.5.7 Random Subspace

The Random Subspace classifier is an ensemble learning method that functions in a similar way to Bagging. It differs in its functionality that it fills the subsets with features instead of instances. Random Subspace functions are best being a base classifier (Ho, 1998). The parameters used for Random Subspace are numIterations set to 10, seed set to 1 and SubSpaceSize set to 0.5.

3.6.5.8 REPtree

Reduced Error Pruning Tree (REPtree) classifier builds multiples trees. It uses regression tree logic by creating many trees in different iterations. Then it choses the best tree from all the generated ones and it will be the representative. The tree is pruned using reduced error pruning (Jayanthi at el., 2013). The parameters used for REPtree are maxDepth set to -1, minNum set to 2.0, numFolds set to 3 and seed set to 1.

3.6.5.9 Decision Table

In machine learning, the simplest way to represent the output is by making it the same as the input and this process is handled using a table. When a decision table is created, some of the attributes can be selected while others will be discarded (Witten I. at el. 2011). The part where the decision is taken as to which of the attributes can be discard without effecting the final decision is important.

Cross validation is used to evaluate all subsets to identify the best one (Kohavi, 1995). The parameter used for Decision Table is search set to BestFirst.

3.6.5.10 M5Rules

M5Rules is a tree structure model that uses a separate and conquer approach to produce a decision list for regression problems. In each iteration the M5 model tree is used to build a model and based on it the best leaf is selected and a rule is generated (Holmes at el., 1999). The parameter used for M5Rules is minNumIstances 4.0.

3.6.5.11 Additive Regressions

This classifier adds up several linear regression models to improve the performance. In each iteration, the created model will depend on the results of the previous model generated. During the time of prediction, the output of all the models will be combined to generate the final prediction (Witten I. at el. 2011). The parameters numIterations is set to 10 and shrinkage is set to 1.0.

3.6.5.12 Decision Stump

This is a simple, one level decision tree classifier. The decision is made based on one input feature. This algorithm is usually used as a base classifier for ensemble learning algorithms (Witten I. at el. 2011). The parameter used for Decision Stump is debug and its set to true.

3.7 The Euler Software

To capture the readings from the different sensors, there is a need to have a handheld device with an application that would capture the raw data gathered by the sensors embedded within the device. During the first stage of experiments conducted within this thesis a Nexus 7 tablet PC (7", version 2013) was used, which is manufactured by Asus. It consists of Ambient Light Sensor, Gyroscope and an Accelerometer that were required for the experiments to be conducted. During the latter stage of research conducted, it was decided to use a mobile phone device for the rest of the experiments conducted within the thesis since most users use mobile phones than tablets. A Samsung (Galaxy S4) was chosen as the capture device it had all the required sensors.

To enable capturing the sensor readings, a software application named 'Euler' was designed and implemented in collaboration with the industry collaborator of this research project. The application enables capturing the illumination values read by the ALS and the values of the three axes of the Accelerometer and of the Gyroscope. The application has a start and a stop button to enable starting up the application to capture raw data and to stop it whenever it is needed. The readings of the different sensors were captured at every 100ms interval, i.e. 10 values were captured per second.

The processes followed within the software application framework can be described as follows:

1) Create a SensorManager class obtained from system sensor service.
2) Get the required Sensor class object from the SensorManager.
3) Create a SensorEventListener interface for the application.
4) Register SensorEventListener to the SensorManager class.
5) After registration, the application will receive SensorEvent object in the onSensorChanged() method callback of SensorEventListener.
6) The application unregisters the SensorEventListener from the SensorManager class when it does not need the data of the sensor anymore.

The sensor framework consists of four main components namely SensorManager, SensorEventListener, Sensor and SensorEvent. SensorManager class provides access through the application to the physical or synthesis sensors. It allows the application to request the information of the sensor and register it to enable receiving its data. SensorEventListener interface provides OnSensorChange() and OnAccuracyChange() callbacks. If there is change in the sensors value OnSensorChange() is called. OnAccuracyChange() is called when the accuracy of the sensor values is changed. Third component is the sensor class, which provides several options to enable identifying the sensors capabilities. Fourth and last is SensorEvent, reports the changes that happen in

the sensor values to the EventSensorListener. This class is used by the operating system.

3.8 Conclusion

This chapter has introduced the reader to the fundamentals of the theories, concepts and processes that are used in the contributory chapters, i.e. Chapters 4-7. In particular the chapter has presented the mathematical models used for illumination modelling, details of various sensors available within a mobile handheld device, machine learning algorithms that can be used to model illumination and also predict device behaviour using sensor data fusion and various evaluation metrics that can be used to determine the accuracy of different learning algorithms. The chapter provides a foundation to the novel knowledge generated within the subsequent chapters of this thesis.

Chapter 4 : Sensor Fusion for Activity Recognition

4.1 Introduction

The present generation of handheld mobile devices comes equipped with a large number of in-built sensors. The key sensors include the Ambient Light Sensor, Proximity Sensor, Gyroscope and the Accelerometer. Many mobile applications are driven based on the readings obtained from either one or more of these sensors. The presence of multiple-sensors will enable the determination of more detailed activities that are carried out by the bearer/user of a mobile device, thus enabling smarter mobile applications to be developed that responds more appropriately to user behaviour and device usage. This way, the user experience can be improved significantly.

Traditionally machine learning algorithms (see section 3.4.3 in chapter 2) have been used to learn and model the behaviour of multivariate data, where the models have been used to predict data elements based on other data known to have an impact on them (see section 2.2 in chapter 2). Thus for determining the behaviour of a mobile handheld device, algorithms from recent advances in machine learning can be used to fuse together the data obtained from the various sensors of a mobile device. In particular the use of bagging and other ensemble classifier based approaches can be proposed to identify a mobile device's behaviour in the space it is present. As the sensor readings are noisy and may include a proportion of missing values and outliers, filtering approaches will have to be used to clean the raw data obtained from the sensors, before their use.

In chapter 2 it was revealed that in the area of activity recognition of mobile handheld devices, different researchers have used different statistical and machine learning techniques to identify the activity that is performed by the user. In most previous work, only the accelerometer was used in activity recognition. However in few cases the data gathered from both the accelerometer and gyroscope was used for activity recognition. In all work the machine learning models tested were rather limited and no rigorous

experiments were conducted to analyses the performance of different machine learning algorithms in their ability to model the behaviour. Further, in some cases, instead of using the raw signal data for learning, an intermediate stage that performed a feature extraction from the raw data was used. However such attempts increase the computational cost of the associated process and may result in the overall process being not suitable for use within computationally constraints application domains such as in mobile handheld devices.

In the approach proposed in this chapter we base the behaviour model building on the raw data gathered from both the accelerometer and the gyroscope. Further a comprehensive and an in-depth analysis has been carried out to compare and contrast the performance of different machine learning algorithms, including ensemble-learning algorithms, in activity recognition.

For clarity of presentation this chapter has been divided into several sections. Section 4.2 presents the experiments conducted in support of the research presented in this chapter. Section 4.5 concludes the chapter providing a brief summary of findings.

4.2 Experiments

Mobile handheld devices can be used for different purposes (e.g., telephone conversations, accessing the web, watching a video, reading a document, using as a torch etc.), can be kept by a user in different places (e.g., on a table, in the hand, in the pocket, etc.) and be used while engaged in different activities (e.g., while seated, walking, jogging, running, traveling in a car, lying down, etc.). The above complex usages make it very difficult for the designers of mobile devices and Apps to know how to optimize the user experience provided by a device or an App. However the presence of multiple sensors in a modern handheld device opens up the possibility of using the data collected by these sensors to identify the usage and thus accordingly optimize the user experience.

In the research conducted within the scope of this thesis, seven different experiments were designed to gather data from specified sets of scenario of mobile phone usage. Most scenarios represent the most common day to day activities that a user will be involved in, when using a phone. However, the scenarios where the user is intentionally shaking phone while performing the activity are the scenarios that are less likely to happen in real life. They had to be tested to enable developers to identify and consider when the user is not serious in looking in the data presented on the screen. Although, as described in Chapter-3, the sensor data gathering App is able to capture data from multiple sensors, the focus in these experiments were on the data captured by only the accelerometer and gyroscope sensors.

Two different people repeated each of the seven scenarios five times to gather data from the two sensors. In all the scenarios the user was holding the phone in his/her hand and performing the activity. These scenarios are:

1) The user sits down on a chair, holds the phone still with his/her right hand at about the chest level, at an inclination angle of 45 and looks at the screen. (Sit)
2) The user sits down and shakes the handheld device firmly in different random directions, with no intention to read or look at the content. (Sit-Shake)
3) The user stands up and starts walking at a normal speed while holding the phone at chest level as if he/she is reading something from the phone. (walk)
4) The user carry's on walking, looks at the screen content and starts to shake the handheld device randomly with no intention of reading the content. (walk-shake)
5) The user starts running and looks at the screen occasionally trying to read. (run)
6) The user carry's on running and shakes the device randomly with no intention of reading the content. (run-shake)
7) The user stands still and looks at the screen. (stand)

The seven scenarios were carried out in sequence, with no gaps in time between them, allowing each scenario to be performed for few seconds. Before the start of the experiment, the application starts data collection and once the last experiment is over, the user stops data collection.

4.2.1 Data Capturing Application

The data capture App was installed in a Samsung galaxy S4 mobile phone. The captured data was stored in a .CSV file and later transferred to a PC for further analysis using Machine Learning algorithms. Note that both the accelerometer and gyroscope sensors produce three values each, being their corresponding readings in the X, Y and Z directions.

Table 4.1 illustrates a sample of the data values captured from the accelerometer and gyroscope sensors. One of the seven attributes presented in table 4.1, were noted as a class label for each instance in time. Note that the seven scenarios were given the names, sit, sit-shake, walk, walk-shake, run, run-shake and stand. The size of the data is 94, 95, 80, 95, 83, 92 and 78 for the activities sit, sitshake, walk, walkshake, run, runshake and stand respectively.

Table 4.1 A sample of data captured from the Accelerometer and Gyroscope sensors

Accelerometer X-axis	Accelerometer Y-axis	Accelerometer Z-axis	Gyroscope X-axis	Gyroscope Y-axis	Gyroscope Z-axis	Class
.....	.....	.....	.....	.....	.....	.....
-0.6568	4.4458	9.0075	-0.0329	-0.0032	-0.0408	sit
-0.6816	4.443	9.0006	-0.033	0.0036	-0.0306	sit
-0.7063	4.4402	8.9938	-0.0332	0.0103	-0.0205	sitshake
-0.7081	4.435	9.0098	-0.0386	0.0117	-0.0175	sitshake
.....	.....	.....	.....	.....	.....	.....
6.7259	0.6587	8.278	1.6328	-0.5285	-0.7291	sitshake
4.8091	-0.914	9.0906	0.9244	-0.38	-0.4849	sitshake
2.9623	-2.3579	9.7293	0.3238	-0.3665	-0.1558	sit
2.3082	0.1208	9.337	0.1646	0.0577	-0.8813	sit
.....	.....	.....	.....	.....	.....	.....
-0.257	3.8822	9.1348	0.0103	-0.0723	-0.527	sit
-0.2969	4.0724	9.1453	0.0146	-0.154	-0.6235	sit
-0.0997	4.0437	9.1012	0.0648	-0.1902	-0.5889	walk
-0.1811	4.0106	9.1403	0.0288	-0.2018	-0.6134	walk
.....	.....	.....	.....	.....	.....	.....
-0.3309	3.6826	8.9728	0.0909	-0.0903	0.0347	walk
-0.204	3.8322	8.9227	0.163	-0.0673	0.0825	walk
-0.108	4.0688	9.0682	0.1748	-0.1102	0.2297	walkshake
0.2403	3.8048	9.1165	0.1957	-0.1096	0.3134	walkshake
.....	.....	.....	.....	.....	.....	.....
3.8879	-3.9017	5.4209	-0.0607	0.6754	-2.8211	walkshake
5.9371	-1.7883	6.5894	-0.1079	0.7281	-1.6559	walkshake
8.2076	0.5776	5.5246	-0.3642	0.9944	-0.7279	run
6.5555	-0.979	4.535	-0.6581	0.9614	-0.397	run
.....	.....	.....	.....	.....	.....	.....
0.3803	5.0907	4.949	-0.2044	0.1606	0.6154	run
1.305	4.3135	3.9933	-0.1818	0.2764	0.6987	run
1.9073	4.7117	5.5778	-0.4261	0.2096	0.5593	runshake
2.2446	4.6878	5.5778	-0.6096	0.0584	0.2849	runshake
.....	.....	.....	.....	.....	.....	.....
-1.3922	-0.7977	5.0257	0.6047	0.5588	-0.4865	runshake
0.184	1.2885	5.0758	1.2551	0.3532	0.4035	runshake
-1.8496	0.0676	8.2132	0.9045	0.1234	-0.4697	stand
0.0939	2.437	7.8257	0.4447	-0.0577	-1.2366	stand
.....	.....	.....	.....	.....	.....	.....

4.2.2 Experimental Results and Analysis

Figure 4.1 plots the x, y, z direction gyroscope and accelerometer readings against time. A detailed visual analysis of figure 4.1 reveals that the relative magnitudes of the three sub readings of the accelerometer and the gyroscope could be used for event classification.

For example from point 1 to point 97, the user was sitting holding the device still. This is indicated by the very low gyroscope values due to lack of translational movement in the phone. Only the X and Z components of the accelerometer indicates any significant value. Whilst Z direction accelerometer reading indicates the gravitational acceleration the phone is subjected to (approximately 9 in magnitude), as it is held at an angle to enable viewing the device's screen, the X direction accelerometer reading indicates the gravitational acceleration component in that direction.

The events that occur between the points marked 97 and 193, 449 and 543, 641 and 737, represent the scenarios where the device was shaken. This is indicated by the relatively and slightly higher values of the gyroscope readings in all directions (due to physical, limited, translational movement) as compared to instances where the phone was held still and somewhat different readings obtained from the accelerometer due to the impact the limited translational motion the phone is subjected to if the user is standing and shaking the phone, a slightly higher translational movement the device is subjected to if the user is walking and shaking the device. In particular between points indicated by 385 to 449, the user was walking indicated by the somewhat changing/noisy nature of the accelerometer and gyroscope readings.

A close review of the observations above indicates that coming up with manually defined thresholds for the six sensor readings that will allow the classification of the seven events will be still a tedious task. This was the approach adopted by the industrial client of this research work, when setting up a suitable logic for the activity recognition based on the captured sensor readings.

Further the raw signals captured by the sensors are noisy and will therefore may to be cleaned prior to further analysis. However a further investigation is required to ascertain whether the level of noise added has an impact on

classification as if it is not the case noise filtering will not be required. Below we provide a further investigation to obtain an answer to this research question.

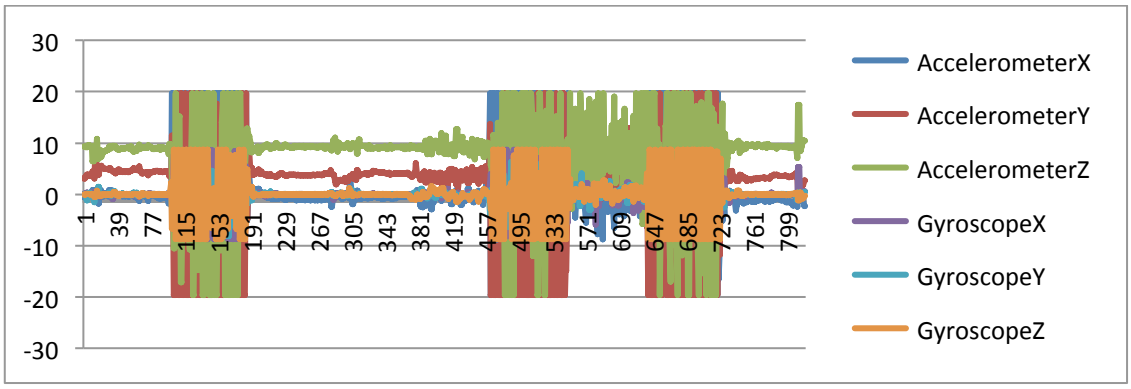
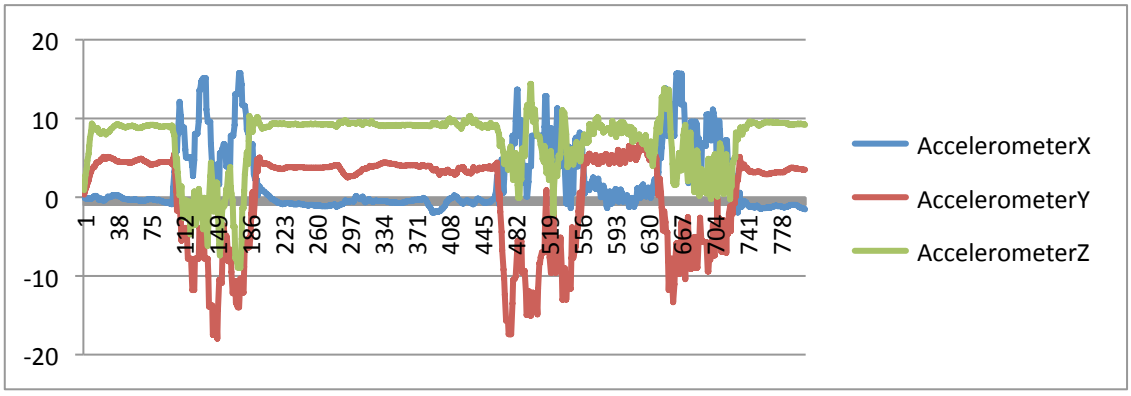
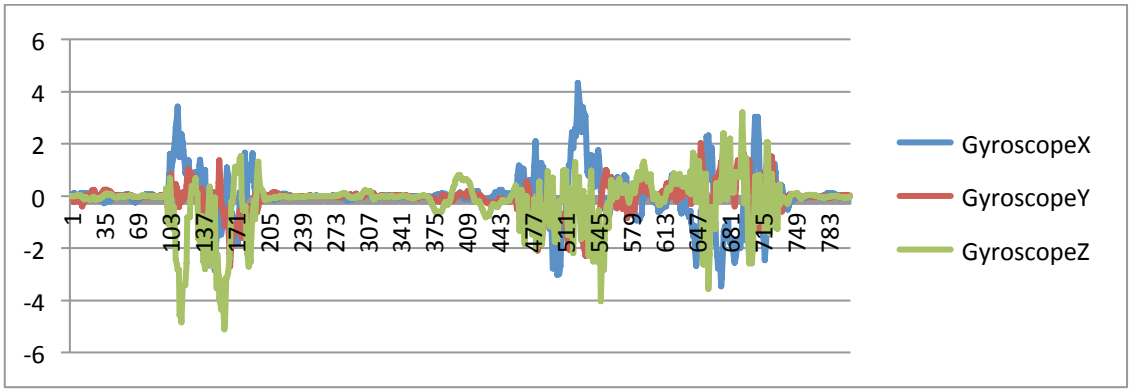


Figure 4.1 Captured Raw Accelerometer and Gyroscope Data

To smoothen the data and minimize noise from the captured raw data of both sensors, a Moving Average Filter was used with the length 10. Figure 4.2 illustrates the filtered signals of the accelerometer and gyroscope readings. Both signals now have a significantly reduced amount of noise and thus should be ready for further processing, i.e. for event classification. The classification done with and without noise removal will indicate the impact that noise has on classification accuracy (see figure 4.3, section 4.2.3).



(a) Accelerometer data after noise reduction



(a) Gyroscope data after noise reduction

Figure 4.2 Accelerometer and Gyroscope Data after using a Moving Average Filter for Noise

4.2.3 Classification

For the purpose of classification of events four classification algorithms were used as implemented within WEKA (Waikato, n.d.) Data mining tool. These are Multi-Layer Perceptron (MLP), Random Forest, and Bagging. It is noted that MLP is a single layer classifier whereas Random Forest and Bagging are ensemble classifiers. As bagging involves the use of a base classifier, it was experimented with two base classifiers, namely Random Forest and MLP. Bagging is said to be effective and simple in reducing the error rate for many learning algorithms (Perdo Domingos, 1997). Based on the work of ((Dietterich, 1990), ensemble learning has been proven to be more accurate as compared to is better than single classifier.

4.2.3.1 The Impact of Noise Removal

In order to investigate the impact of noise reduction in overall prediction accuracy of learning algorithms, figure 3.3 illustrates graphs that compare the performance of the four classifiers when the moving average filter was and was not used on the raw data captured from the two sensors. The vertical axis depicts the correctly classified percentage of all events tested. The biggest impact on noise reduction has been when the MLP was used as a single classifier and when used within the ensemble classifier, bagging. This is indicated by the

15% approximate performance improvement. No significant improvement has been obtained by noise removal when using the Random Forest algorithms as a classifier, indicating its robustness to the presence of noise.

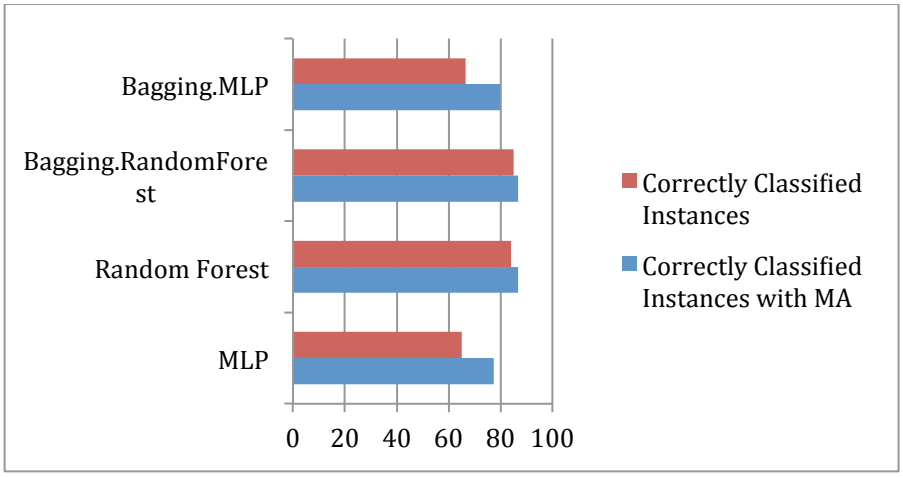


Figure 4.3 Performance comparison with and without noise reduction - % correct total event predictions

4.2.3.2 The Impact of Fusing the Accelerometer & Gyroscope Readings

Table 3.2, illustrates the ability of the four classifiers to identify the seven events based on the individual classifier readings, i.e. the accelerometer and gyroscope readings considered independently and individually. The percentage accuracies of the predictions are tabulated. The highest accuracy is obtained for identification of sitting with all classification approaches indicating accuracies of over 90%. Running can also be classified relatively accurately. When the sensors are considered independently it is the less complex tasks that can be identified more accurately. This is well indicated by the dismal performance of the classifiers when identifying events that includes shaking as an added complication. Further observation is that the MLP as a single classifier has the worst average performance and bagging with random forest as the base classifier has the highest accuracy of performance.

Table 4.2 Percentage correctly predicted instances when using the Accelerometer and Gyroscope, independently

	Correctly Predicted Instances (%)							
	When using Accelerometer				When using Gyroscope			
	MLP	Random Forest	Bagging. MLP	Bagging. Random Forest	MLP	Random Forest	Bagging. MLP	Bagging. Random Forest
Sit	92	96	91	95	95	94	95	94
SitShake	34	70	34	59	25	73	38	67
Walk	18	86	21	80	6	78	10	72
WalkShake	52	64	53	62	42	84	48	78
Run	77	86	85	88	49	83	49	83
RunShake	32	55	28	55	51	75	54	73
Stand	50	74	51	71	0	69	0	60
Overall	60	80	61	78	53	83	56	80

Table 4.3 illustrates the percentage of the correctly predicted instances when the readings of the two sensors were combined and used in association with each of the four classification algorithms. A significant improvement in overall accuracy is clearly noticeable when results of table 4.3 are compared with those tabulated in table 4.2. The more common and simpler activities such as sitting, walking, running and standing, indicates accuracy levels of over 80% when using all classification approaches. Sitting seems to be the easiest to be identified amongst the rest. The overall percentage of the correctly predicted instances is the same for the algorithms, Random Forest and Bagging with Random Forests when classifying simpler tasks such as sitting, standing, walking and running. However in classifying more complex scenarios that involve shaking the two said classifier performance is mixed. However, overall bagging with random forest provides the best classification accuracy. MLP on the other hand, as a non-ensemble classifier, has generally poor accuracy results, when compared to the rest of the models investigated. Even though when MLP is used as the base classifier within

Bagging the accuracy shows slight improvement it is still worse when compared to the accuracy obtained from the random forest based approaches.

When comparing the results tabulated in Table 4.2 and 4.3, when the data from the sensors were combined, the overall results for Bagging with Random Forests has increased from 80% to 86%. The accuracy in using Bagging with MLP for classification indicates a greater improvement in accuracy from 60% to 80%. Random Forests have generally given the same accuracy results of above 80%. Finally the accuracy obtainable with MLP as a single classifier increased from 60% up to 77%. In summary it can be concluded that when the two sensor readings are combined the models can predict with higher rates of accuracy.

Table 4.3 Accuracy of the Correctly Predicted Activities when combining Accelerometer and Gyroscope data

	Correctly Predicted Instances (%)			
	MLP	Random Forest	Bagging. MLP	Bagging. Random Forest
Sit	87	94	90	95
SitShake	73	86	78	77
Walk	46	92	48	92
WalkShake	75	82	85	80
Run	83	80	88	88
RunShake	73	70	70	75
Stand	76	83	70	83
Overall	77	86	79	86

4.2.3.3 Impact of Using Different Classification Algorithms

Table 3.4 illustrates the confusion matrix obtained when MLP and Random Forest are used a single classifiers. It indicates the presence of prediction errors specifically when recognising certain scenarios, more than some others. When using the Random Forest classifier, 14 known Run-Shake scenarios were classified wrongly as Walk-Shake. 8 Walk-Shake cases have been incorrectly predicted as Sit-Shake. There was a slight confusion between the Stand and Sit scenarios where 6 occurrences of Stand have been wrongly classified as Sit.

The predicted class for MLP had more incorrect predictions compared to Random Forests. Major inaccuracies of classification were with Walk as it was incorrectly predicted as Sit, 22 times and as Stand, 13 times. Similar to the case of using Random Forests, there was confusion between identifying Walk-Shake and Run-Shake scenarios as both activities include shaking the phone. It was also slightly harder to identify if the user was walking or running. In conclusion, Random Forests gives an overall better result as compared to MLP.

Table 4.4 Confusion Matrix for Random Forests and MultilayerPerceptron

		Predicted Class Random Forest							Predicted Class MultilayerPerceptron						
Actual Class		Sit	Sit Shake	Walk	Walk Shake	Run	Run Shake	Stand	Sit	Sit Shake	Walk	Walk Shake	Run	Run Shake	Stand
	Sit	266	0	4	2	7	0	2	247	0	5	6	9	2	12
	Sit Shake	3	84	0	7	1	2	0	8	71	1	10	1	6	0
	Walk	6	0	74	0	0	0	0	22	0	37	0	0	8	13
	Walk Shake	2	8	1	78	1	4	1	2	11	0	72	2	7	1
	Run	9	1	0	2	67	4	0	8	0	2	0	69	4	0
	Run Shake	3	5	0	14	4	65	1	0	7	1	12	4	68	0
	Stand	6	1	3	1	2	0	65	11	0	5	1	1	0	60

Table 4.4 illustrates the Confusion Matrices when using Bagging with base classifier Random Forests and MLP. Bagging with Random Forests has given its best prediction in recognising Walking as it was mistakenly predicted as only Sitting, 6 times. It was not confused with any other activity. The possibility of this confusion is during the sitting experiment, the user was not sitting still but was performing minor natural movements while holding the phone. The same prediction accuracy was indicated for Walking activity when using Random Forests classifier as a single classifier (see Table 4.3). When the phone was shaken, there were more incorrect predictions as compared to when it was not shaken in carrying out the simpler activities on their own.

When using Bagging with MLP as the base classifier, there were a noticeable number of incorrectly classified walking activities. It has been confused 25 times with siting and 13 times with standing. Run-Shake was incorrectly predicted 10 times as Sit-Shake and 11 times as Walk-Shake. The common activities show a large number of incorrectly classified instances as compared to when Random Forest was used with Bagging for prediction. The activity with the least number of classification errors is Run activity. However, Bagging with MLP as a base classifier has given better overall accuracy results when compared to MLP used as a single classifier.

Table 4.5 Confusion Matrix for Bagging with MLP and Random Forests as the base classifiers

		Predicted Class Bagging with MLP as base classifier							Predicted Class Bagging with Random Forest as base classifier						
Actual Class		Sit	Sit Shake	Walk	Walk Shake	Run	Run Shake	Stand	Sit	Sit Shake	Walk	Walk Shake	Run	Run Shake	Stand
	Sit	255	2	6	4	6	2	6	268	0	2	2	6	0	3
	Sit Shake	8	76	0	7	2	4	0	7	75	0	9	2	4	0
	Walk	25	0	39	0	1	2	13	6	0	74	0	0	0	0
	Walk Shake	3	6	0	81	1	4	0	3	6	2	76	2	5	1
	Run	7	0	1	0	73	2	0	6	1	0	0	73	3	0
	Run Shake	0	10	0	11	5	65	1	2	3	0	15	3	69	0
	Stand	14	0	6	2	1	0	55	5	0	6	1	1	0	65

Table 4.5 compares the performance of using Bagging with Random Forests and the J48 classifier. Both confusion matrices tabulated are similar with a very slight difference in their values. In the experiments conducted within the scope of this chapter, in general, the Bagging classifier with Random Forest as the base classifier has given the best results amongst all of the algorithms. However, implementing it on handheld devices is complicated due to the nature of its functionality by generating a forest made up of a large number of trees (See 3.6.5.3). Using J48 classifier on the other hand, is less complicated to implement, as it is a single tree structured algorithm (see 3.6.5.4). The results in Table 4.5

illustrates that it has reduced the correctly classified instances to approximately 3% compared to bagging.

Table 4.6 Confusion Matrix for Bagging with MLP and J48 classifiers

		Predicted Class Bagging with Random Forest as base classifier							Predicted Class J48						
Actual Class		Sit	Sit Shake	Walk	Walk Shake	Run	Run Shake	stand	Sit	Sit Shake	Walk	Walk Shake	Run	Run Shake	stand
	Sit	267	0	4	3	4	0	3	261	3	3	1	7	2	4
	Sit Shake	6	76	0	10	2	3	0	1	75	0	8	4	8	1
	Walk	9	0	70	0	0	0	0	2	0	76	0	0	0	2
	Walk Shake	3	5	0	77	2	7	1	3	8	0	70	1	13	0
	Run	5	0	0	0	74	4	0	12	0	1	1	61	5	3
	Run Shake	1	4	1	12	4	69	1	1	10	1	11	7	62	0
	stand	4	0	7	0	1	1	65	4	0	4	1	3	0	66

4.2.3.4 Classification based on Sensor Value Change

In the previous experiment the classification was based on data captured when the user was clearly performing a given task, out of the seven scenarios. The data when the user changes the activity from one task to another was not the focus. In this experiment data gathered when the user was changing activity (i.e. the boundary data), e.g. when the user was initially sitting and then starts walking, was considered. Seven boundary conditions were considered as named in table 4.6. Only K Nearest Neighbour was considered for prediction as this was proven to be the most accurate and stable learning algorithm for activity recognition. The results are tabulated in table 4.6. It is noted that the table tabulates results when the individual sensors were separately used and combined.

The results tabulated in table 4.6, indicates that if K Nearest Neighbour is applied on the data captured from the accelerometer and gyroscope sensors independently the percentage of the correctly predicted instances are lower than when they are combined. The overall percentage of the correctly predicted instances for accelerometer and gyroscope data combined is 85.

Table 4.7 Correctly predicted instances when the change from one activity to another is considered

	Correctly Predicted Instances (%)		
	Accelerometer values	Gyroscope values	Accelerometer and Gyroscope values
	K Nearest Neighbour		
Sit to SitShake	76	70	90
SitShake to Walk	60	45	85
Sit to Walk	80	90	95
Walk to WalkShake	65	50	75
WalkShake to Run	65	70	90
Run to RunShake	70	60	95
RunShake to Stand	40	50	50
Overall	67	63	85

4.2.3.5 Performance Comparison with Approaches used in Literature

In the activity recognition area, researchers have used a number of Machine Learning algorithms to identify between the different activities performed by the user. (Awan et al., 2012) have proven that Naïve Bayes is the best algorithm for activity recognition. (Anjum & Ilyas, 2013) on the other had proved that J48 performed better than Naïve Bayes and K-NN. (Kwapisz et al., 2011) proved that MLP gives a higher accuracy than J48. In 2015, (Coskun et al., 2015) use just Random Forest algorithms as it has been successfully used in activity recognition field.

One of the most commonly used classifiers for activity recognition is KNN (K-Nearest Neighbour) and decision tree classifiers (Coskun et al., 2015). Consequently it was decide to include these classifiers and compare their performances with the results of algorithms experimented previously within the research context of this chapter. It is noted that in majority of the above work where KNN, J48 and Naïve Bayes algorithms were recommended. The research only made use of the accelerometer data, i.e., not combined with gyroscope data. Understanding the performance of all algorithms when tested on combined data will be useful contribution to existing research. Table 4.8 tabulates the results.

Table 4.8 Comparison of correctly predicted instances between different classifiers

	Correctly Predicted Instances (%)							
	MLP	Random Forest	Bagging. MLP	Bagging. Random Forest	K-Nearest Neighbour	J48	Naïve	Logistic Regression
Sit	87	94	90	95	97	92	90	96
Sit Shake	73	86	78	77	84	77	76	64
Walk	46	92	48	92	96	95	83	16
Walk Shake	75	82	85	80	86	73	78	76
Run	83	80	88	88	89	73	85	65
Run Shake	73	70	70	75	80	67	72	67
Stand	76	83	70	83	92	84	64	64
Overall	77	86	79	86	91	83	80	72

Table 3.6 illustrates that overall the highest accuracy is of the KNN classifier giving an accuracy of 91%, followed by Bagging with and without Random Forest with overall accuracy of 86%. KNN has outperformed all other tested classifiers when the user is sitting, walking, running and standing. The time the system was fooled by shaking the phone during the activities, Random Forest has given a higher accuracy in SitShake activity recognition, while K-NN gives the highest accuracy in the recognition of rest of the shake activities. In general, the highest identifiable activity is sitting then after it comes walking on the second place then running.

Neural network classifier has given the least accuracy amongst the classifiers tested after Logistic Regression that has given an accuracy rate 5% less than MLP. Tree based classifiers has given a higher accuracy rate in general compared

to Neural Networks. However, the K-NN classifier gives the highest accuracy overall.

4.2.3.6 Optimization

In section 3.5, all the classifiers where processed using their default parameter settings as presented in weka. Some of the used classifiers were taken and optimization was done to them. Optimization is where you find the best setting of the parameters, which can increase the performance of the model. Table 3.7 illustrates the results.

Table 4.9 Few algorithms of table 4.7 optimized

Algorithm	Accuracy	Parameter Changed	Old value	New value	New Accuracy
MLP	77.4194 %	learningRate	0.3	0.1	79.4045 %
		momentum	0.2	0.1	
		trainingTime	500	700	
Random Forest	86.2134 %	numFeatures	0	2	89.4541%
		numTrees	100	75	
K- Nearest Neighbour	91.067%	K value	1	3	89.33%

The results in table 3.7 illustrates that there is a minor increment in the accuracy rate. This indicates that change in the parameters of different chosen algorithms in this set of data dose not show much of a difference. Consequently, it was decided to use the default setting of the parameters.

4.5 Conclusion and Future Work

In this chapter the possibility of activity analysis and recognition in relation to the use of a mobile phone, based on fusing the data captured from multiple sensors was investigated. A proof of concept that machine-learning algorithms can be utilized for this purpose has been provided based on fusing the data captured from the accelerometer and the gyroscope of a handheld device.

The use of different well-known machine learning algorithms has been investigated in detail for the classification of seven common events associated with the typical use of a mobile device. It has been shown that the K-NN classifier performs best giving accuracy figures of beyond 90% in identifying all events tested.

The ability of the ensemble learning algorithms to accurately identify events when the scenarios increase in complexity has been demonstrated by the research conducted in this chapter. Bagging is found to be the second most accurate amongst the algorithms compared while K-NN has proven to be most accurate amongst them all. However, when it comes to implementation, J48 is much simpler to implement. Therefore considering the efficiency vs. the implementation complexity, the J48 classifier was chosen as the algorithm for implementation on a mobile device environment. J48 have provided accuracy of 83% and in terms of implementation it is a simple tree based algorithm, which makes implementing easier than random forest.

It is noted that the experiments conducted within this chapter made use of default parameter values, when making use of each machine-learning algorithm, for activity recognition. It is possible to optimise the performance of each algorithm by making the best selection of parameters. Although parameter based optimisation algorithms could have been used our preliminary research in this direction did not indicate any significant improvement of accuracy for all algorithms used.

It is also noted that the activity recognition was carried out completely based on raw data, on which a level of noise removal was conducted based on using a moving average filter. It may be possible to use secondary features extracted from this raw data to feed into the learning algorithms. Although this has been the popular approach in literature, due to the fact that such secondary feature

extraction algorithms needing additional compute power, within the context of research conducted in this chapter, it was decided to work based on the raw data. Our preliminary results indicated that feature extraction prior to training and using such features in training the learning algorithms only have a marginal impact on the accuracy of the classification capabilities.

Having proven that the various sensors present in a mobile device can be effectively used for mobile device activity recognition, Chapters 5 and 6 present approaches to mathematical and machine learning based modelling of illumination within a scene in which a mobile phone is kept. In chapter-7 the activity recognition research presented in this chapter was used to improve the illumination modelling approaches presented in Chapter-7, by designing an illumination model that is sensitive to the usage of the mobile device and can accurately predict illumination in a three dimensional space.

Chapter 5 : Modelling of Illumination

5.1 Introduction

Overview chapter discussed the need for illumination modelling of an environment in which a mobile phone is present to overcome the limitations of Ambient Light Sensors, thus making the device and it's App's quickly adoptable to environmental illumination changes. This way the mobile handheld device can provide a more positive user experience.

This chapter presents an investigation of the mathematical formula that can be used to model the illumination variations of an environment in which a mobile phone is present. Within the research context of this chapter only the modelling of illumination due to one and two light sources are carried out. The research presented provide a basic yet rigorous understanding of the behaviour of illumination changes that results from the simplest forms of artificially created illumination and optical phenomena it creates in a given environment. This work is subsequently further extended and generalized in chapter 6 to cover an environment that is lit up by evenly spaced multiple light sources, including both point light sources and ambient light. Chapter 7 further extends the work to model illumination in an environment with multiple light sources, placed with random spacing.

For the modelling of illumination as described above, six different mathematical models were implemented and the results were compared and analysed in this chapter. The selection of these specific mathematical models was driven by the potential accuracy some models are likely to provide and the potential simplicity of other models. Whilst accurate models provide a more effective theoretical solution to the latency problem on an ALS their implementation within a mobile device can be restrictive due to the demands on such a device's computational resources/power. In contrast the simpler models, provided being accurate within practical limits, can provide a more implementable solution within a mobile device.

5.2 The Research Problem

The present generation of Ambient Light Sensors (ALS) of a mobile handheld device suffers from two practical shortcomings. The ALS's are narrow angled, i.e. they respond effectively only within a narrow angle of operation and there is a latency of operation. As a result mobile applications that are operated and/or based on the ALS readings could perform sub-optimally, especially when operated in environments with non-uniform illumination. The device responds slowly and intermittently to illumination changes and location of light sources. Thus the applications (i.e. mobile Apps) in which the operations depend on the ALS readings will either adopt with unacceptable levels of latency or/and may demonstrate a discrete nature of operation.

Therefore we propose to come up with a mathematical framework to predict the ambient illumination of an environment in which a mobile device is present. This work is presented in the subsequent sections.

5.3 Modelling Illumination Distribution due to One Light Source

The predictions are based on illumination models that are computed based on a small number of true ALS readings taken during an application calibration stage where the mobile device is kept stationary at a point of measurement for a sufficient amount of time for the ALS to provide a stable and accurate reading. It is noted that the scene is lit by a single point light source, as at this stage the focus is to rigorously study the practical illumination distribution in the environment of a simplest possible light source.

Six different regression models are developed, implemented and compared based on Polynomial, Sine, Sum-of-Sine, Gaussian, Fourier and Smoothing Spline functions. The first three models are chosen due to their computational simplicity and the last three models are chosen based on their potential accuracy. It is shown that the prediction accuracy for all models was found to be about 0.99 when measured using R-Squared test with the best performance

being produced by the Smoothing Spline based on the experimental results presented in table 5.6 and 5.16. Further the chapter discusses the mathematical complexity of each model and investigates how to make compromises in finding the simplest models for an efficient practical implementation as against the best theoretical models.

5.3.1 Design of Experiments

The main purpose of the experiments was to capture actual illumination values at a number of test points spread across the environment (as measured by the ALS when its stable operational point is reached), under one single light source and use these values for determining the best mathematical model that will accurately predict the illumination values anywhere within the said environment.

The experiment was done indoors in a dark room where all the light sources were turned off. A wooden surfaced table of 97.8 cm in length, 67.3 cm in width and 65 cm in height was kept at the centre of the room. A 12 LED light bulb with small surface diameter (i.e. a good assumption of a point light source) with a clear plastic outer cover was used for the experiment; it was securely hung on a tripod at approximately 75 cm height from the table.

For the experiment, a Nexus 7 tablet PC (7", version 2013) manufactured by Asus was used. To enable capturing the ALS sensor readings, a software application named 'Euler' was implemented and developed in collaboration with Apical Imaging Ltd (see section 3.5).

The following six points summaries the steps of capturing the sensor values:

1) All lights in the room were turned off and the door was shut to ensure that the room is dark. Any light entering the room via window panels (only a few were present) were blocked off with the glass panels being sealed off with black paper.

2) The experimental light source was turned on and the Nexus 7 was placed at the end of the table.

3) A 5-pin mini USB cable was connected to the Nexus 7 to enable pulling it towards the other end of the table.

4) The top part of the Nexus 7 where the, ALS is located was placed at the edge of the table and the end where cable connects was placed towards the middle/inside of the table.

5) Once the Euler software was turned on, the Nexus 7 was pulled at a very slow average speed of 0.00014 m/s approximately, towards the end of the table and at this point the software was turned off. The software application reads the ALS values when the device is being moved and records them internally. Due to the slow speed of movement the ALS has time to stabilize recording the stable ALS value at each recording point.

6) Finally the Nexus 7 is connected to the computer to load the captured values of the sensors.

The phone was pulled by hand instead of using a conveyor belt to avoid constant speed. This has resulted in a non-constant speed, which is how the phone is actually moving by the user in a normal case.

A graph is plotted with the actual illumination values placed on the y-axis and time placed on the x-axis. The plot shows a steady increase of the ALS reading towards a peak point of illumination when the phone is kept directly under the light source and then a subsequent decrease towards a lowest point of the illumination values. The resulting graph indicates a distorted bell shape as against a smoother shape that would have been theoretically expected based on the applicability of the well-known inverse square law (O'Nolan John, 2012) in illumination intensity distribution around a point light source.

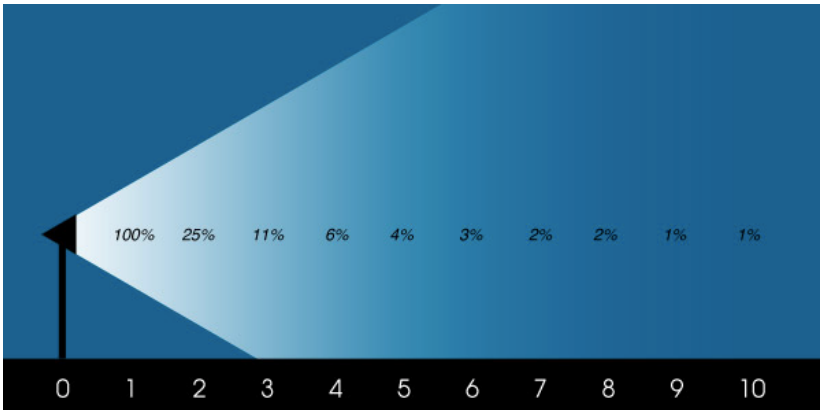


Figure 5.1 Drop of light intensity as the distance increases (O'Nolan John, 2012)

It is noted that in practice, the movement of the phone barely occurs in constant velocity, and strictly in one direction. During the movement of the phone, the distance that the phone has travelled cannot be calculated accurately using the internal sensors, especially indoors where the GPS signals may not be recovered. Consequently in our experiments the time was taken into consideration to replace location/distance against the actual illumination values. The result presented in figure 5.2 indicates that it is more accurate to plot against time since the speed is not constant.

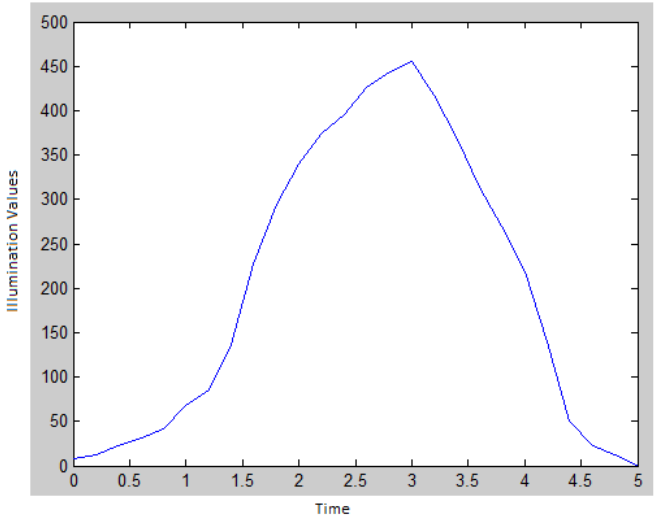


Figure 5.2 The illumination values captured by the ALS under a single light source

A closer investigation of the graph plotted in figure 5.2 reveals that the distortions are due to latency of response of the ALS reader and other would be secondary reflection. An ideal graph should indicate a bell shape given the laws of physics governing light emission from a single point light source.

5.3.2 Results and Analysis

The six mathematical models were investigated in modelling the actual illumination value distributions that were captured in the experiment. This section presents the results originating from applying each of the models and a comparison between them.

5.3.2.1 Sine Function

Three different equations were investigated with respect to modelling the illumination using a single Sine function as follows:

5.3.2.1.1 Equation 1

The first equation that was investigated is $I\ (t) = \ a * sin\ (b * t)$ where a and b represent the amplitude and period, respectively.

Figure 5.3 illustrates the plot of $I\ (t) = \ a * sin\ (b * t)$. It is noticeable that there is a significant difference between the actual illumination values and the values obtained from the plot of the sine function being tested, specifically between x=0 to x=2.5. Beyond x=3, the actual illumination points are more closer to the theoretical values. The shape mismatch that exists between the measured and theoretical graphs suggests the need to shift the mean of the function and hence suggests the addition of a 'constant' term (see the following section).

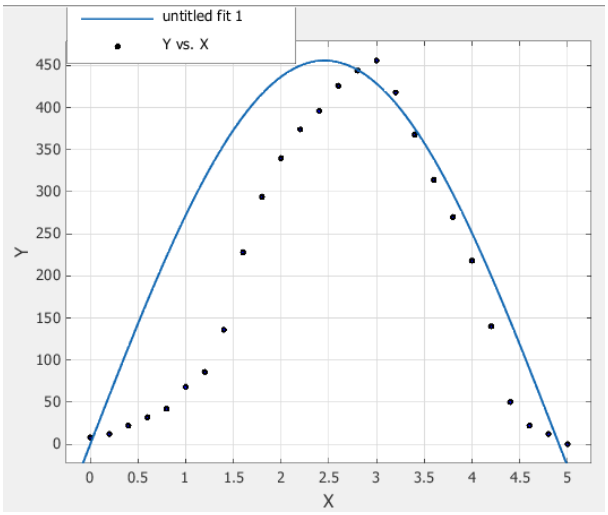


Figure 5.3 Comparison of plots of $I(t) = a * sin(b * t)$ and the actual illumination variation

General model of Figure 5.3:

$$I(t) = a * \sin(b * t) \qquad (4.1)$$

Coefficients:

a = 456

b = 0.6399

Goodness of fit:

SSE: 2.89e+05

R-square: 0.5828

Adjusted R-square: 0.5655

RMSE: 109.7

The result of R-square value is 0.5828 and hence the model is not satisfactory in terms of accuracy of prediction of the actual illumination values. Therefore it was decided to add a constant term to the equation (i.e., shift the mean) and modify it as described in the section that follows.

5.3.2.1.2 Equation 2

The second equation that was investigated is $I\ (t)\ =\ a * sin\ (b * t) + c$ where c is constant that represents the vertical/mean shift of the graph. This was used to

enable shitting the Sine wave vertically to get the best fit to the actual illumination curve.

General model of Figure 5.4:

$$I(t) = a * sin(b * t) + c \qquad (4.2)$$

Coefficients:

a = 499.3

b = 0.5919

c = -124.1

Goodness of fit:

SSE: 1.003e+05

R-square: 0.8552

Adjusted R-square: 0.8426

RMSE: 66.05

It is clearly noticeable that the result of R-Square has improved by 44% as compared to that of the first equation. Figure 5.4 illustrates the actual illumination values and the implementation of equation (4.2).

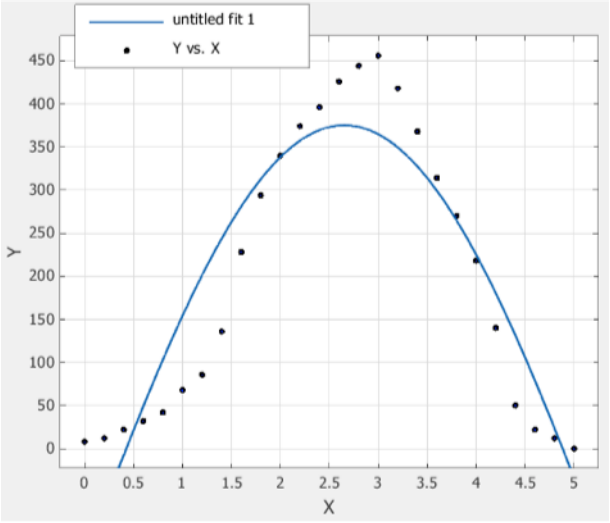


Figure 5.4 Plot of $I(t) = a * sin(b * t) + c$

5.3.2.1.3 Equation 3

Even though the accuracy results obtained above are better than that of the previous experiment, it was decided attempt to increase the prediction accuracy further by taking the power of the sine function. The third equation that is investigated is used $I(t) = a * sin(b * t - y)^n$ where y and n represents the phase shift and exponents, respectively. Using the exponent smoothens the edge of the Sine curve. As a result, the curve will look like a bell shape, which is close to the shape of the actual illumination values. Table 1 compares the goodness of fit for equations with exponents of between 2 to 8.

Table 5.1 Exponents and Goodness of fit of $I(t) = a * sin(b * t - y)^n$

Exponents (n)/ Goodness of fit	2	3	4	5	6	7	8
SSE	7760	7416	8056	8540	8902	9180	9398
R-square	0.9888	0.9893	0.9884	0.9877	0.9871	0.9867	0.9864
Adjusted R-square	0.9878	0.9884	0.9874	0.9866	0.986	0.9856	0.9853
RMSE	18.37	17.96	18.72	19.27	19.67	19.98	20.21

R-Square is in the range of 0.98 in general for all exponents, which is a 17% increase as compared to that of $I(t) = a * sin(b * t) + c$. Table 5.1 tabulates the, SSE, R-Square, Adjusted R-Square and RMSE values and indicates that the exponent 3 has the best fitting values. The graph plotted in figure 5.5 indicates that as the exponent value increases beyond 3, the R-Square value decreases.

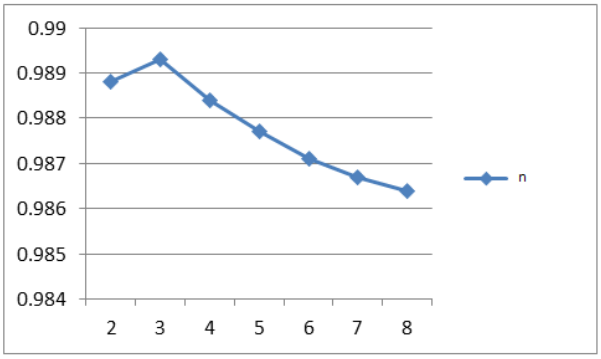


Figure 5.5 Exponents vs. R-Square values of $I(t) = a * sin(b * t - y)^n$

The graph of $I\ (t)\ =\ a * sin\ (b * t - y)^3$ plotted in figure 5.5 illustrates a bell shape as targeted. It is noted that it fits all the points of the true curve, accurately.

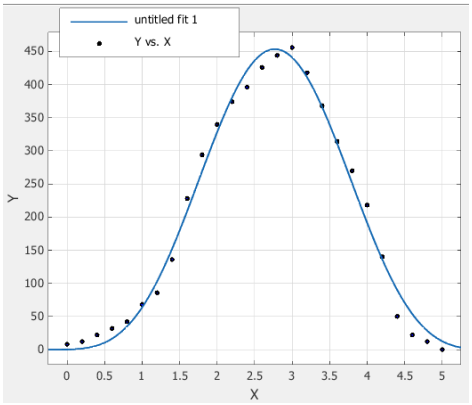


Figure 5.6 Plot of $f\ (x)\ =\ a * sin\ (b * x - t)^3$

General model of Figure 5.5:

$$I(t)\ =\ a * sin(b * t - y)^3$$

Coefficients:

a = 451.5

b = 0.5833

t = 0.04421

5.3.2.2 Polynomial Model

Seeking a better fit, it was decided to investigate the use of Polynomial Models. In the preliminary experimental analysis, 2nd degree quadratic functions and grater were considered. It was observed that higher the degree of the polynomials the corresponding graphs were forming towards a shape of a curve that is close to the actual illumination graph and fitting it with a polynomial was possible with a different level of goodness in each degree level. Consequently, a comparison was done between using 6th, 7th, 8th and 9th degree of polynomials. The 9th degree polynomial gave the best fit in terms of R-Square. However even the 6th degree polynomial gave very accurate prediction results.

Table 5.2 Comparing Goodness of fit for 6th, 7th, 8th and 9th degree of polynomials

Goodness of fit/ Poly Degree	6th	7th	8th	9th
SSE	5617	4990	4945	4943
R-square	0.9937	0.9944	0.9945	0.9945
Adjusted R-square	0.9922	0.9928	0.9926	0.9922
RMSE	14.99	14.42	14.66	14.99

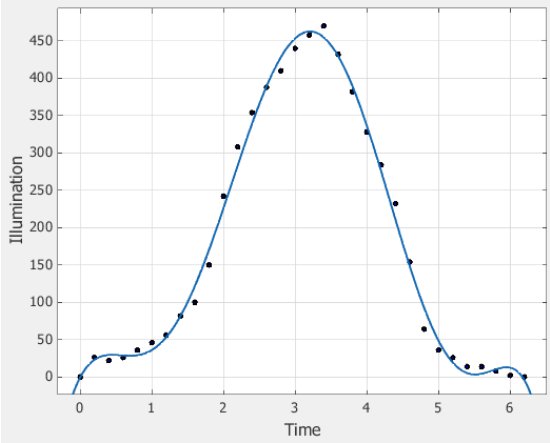


Figure 5.7 Comparison of plots of the actual values and 9th degree polynomial function

General Model of the 9th degree polynomial can be stated as follows:

$$I(t) = p1 * t^9 + p2 * t^8 + p3 * t^7 + p4 * t^6 + p5 * t^5 + p6 * t^4 + p7 * t^3 + p8 * t^2 + p9 * t + p10 \quad (4.5)$$

The Coefficients:

p1 = -0.4593

p2 = 10.19

p3 = -93.73

p4 = 459.4

p5 = -1280

p6 = 1976

p7 = -1524

p8 = 533.6

p9 = -26.82

p10 =7.187

5.3.2.3 Fourier Series

The third mathematical equation investigated is the use of a Fourier series. It has given noticeably good results as indicated by the goodness of fit tabulated in table 5.3. The table shows the goodness of fit obtained by the use of Fourier harmonics from 1st to 8th. It is observed that as the Fourier harmonic increases the overall results of the goodness of fit becomes better.

Table 5.3 Compare Goodness of fit with Fourier series

Goodness of fit/ Fourier harmonic	SSE	R-square	Adjusted R-square	RMSE
1st	1.892e+04	0.9789	0.9767	26
2nd	6236	0.9931	0.9917	15.49
3rd	5337	0.9941	0.9923	14.91
4th	3677	0.9959	0.9942	12.93
5th	2095	0.9977	0.9964	10.24
6th	825.2	0.9991	0.9984	6.771
7th	809.9	0.9991	0.9983	7.115
8th	548.9	0.9994	0.9986	6.262

Looking at table 5.3, the best accuracy is obtained by the 8th Fourier harmonic. It is observed that higher harmonic functions result in higher accuracy.

The equation for the 8th harmonic function is:

$$I(t) = a0 + a1 * \cos(t * w) + b1 * \sin(t * w) + a2 * \cos(2 * t * w) + b2 * \sin(2 * t * w) + a3 * \cos(3 * t * w) + b3 * \sin(3 * t * w) + a4 * cos(4 * t * w) + b4 * sin(4 * t * w) + a5 * cos(5 * t * w) + b5 * sin(5 * t * w) + a6 * cos(6 * t * w) + b6 * sin(6 * t * w) + a7 * cos(7 * t * w) + b7 * sin(7 * t * w) + a8 * cos(8 * t * w) + b8 * sin(8 * t * w) \quad (4.6)$$

The Coefficients are:

a0 = 175.5

a1 = -226.7

b1 = 9.393

a2 = 64.4

b2 = 8.26

a3 = 5.088

b3 = -1.07

a4 = -11.02

b4 = 1.168

a5 = 5.498

b5 = -3.405

a6 = 7.344

b6 = 9.394

a7 = -0.8378

b7 = -2.046

a8 = -4.186

b8 = -0.4183

w = 0.9866

On the other hand, if the implementation of the equation on hardware/software components is to be considered, then the 8th Fourier harmonic will not be the best option due to the need of excessive computations that results in extra hardware/software resource cost. Instead, the 2nd harmonic will be the best choice since the equation is not complex as compared to the 8th harmonic and the accuracy is still very good, R-square value being 0.9931 as against 0.9994.
The equation of the 2nd harmonic equation is,

$$I(t) = a0 + a1 * cos(t * w) + b1 * sin(t * w) + a2 * cos(2 * t * w) + b2 * sin(2 * t * w) \quad (4.7)$$

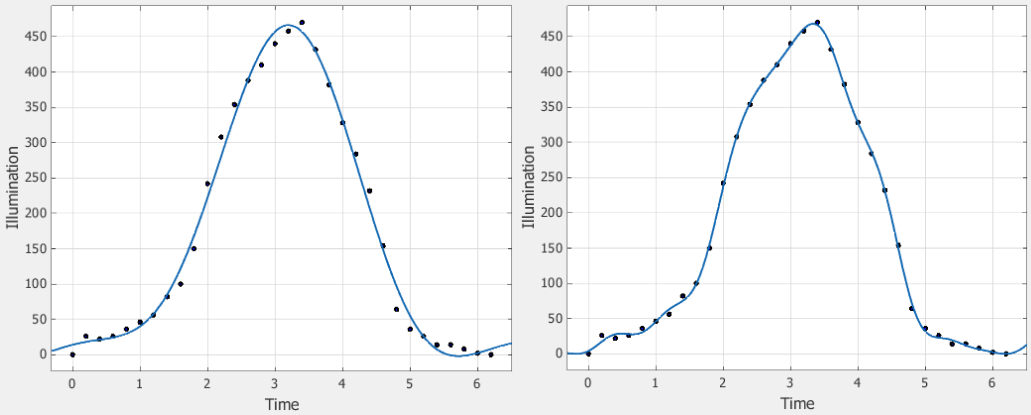


Figure 5.8 Plots of the actual values with 2nd (left) and 8th (right) Fourier harmonics

Figure 5.7 illustrates the 2nd and 8th Fourier harmonic functions plotted against the actual measured illumination values. The 8th Fourier harmonic gives an almost perfect prediction to the actual illumination values. On the other hand, the 2nd Fourier harmonic gives a very close prediction to the actual illumination values. Generally, given the simplicity of the equation, the curve of the 2nd Fourier harmonic is a sufficiently accurate prediction to the actual illumination values in practice.

5.3.2.4 Gaussian Derivative Function

The fourth mathematical equation that was used is the Gaussian derivative function. It uses exponential functions to represent the model. Table 5.4 illustrates the goodness of fit of up to the 5th Gaussian derivative. The 4th derivate has given an accuracy of 0.999 and its overall evaluation matrices points out to it being the best.

Table 5.4 Goodness of fit and Gaussian derivative

Goodness of fit/ Gaussian derivative	SSE	R-square	Adjusted R-square	RMSE
1st	1.061e+04	0.9882	0.9874	19.12
2nd	6116	0.9932	0.9932	15.34
3rd	5349	0.994	0.992	15.25
4th	923.3	0.999	0.999	6.795
5th	3626	0.996	0.996	14.61

The R-Square value of all the derivatives from 2nd up to 5th is above 0.99. This indicates that Gaussian derivatives are a good prediction model within an environment with a single light source.

The equation for the 4th Gaussian derivative is as follows:

$$I(t) = a1 * exp(-((t-b1)/c1)^2) + a2 * exp(-((t-b2)/c2)^2) + a3 * exp(-((t-b3)/c3)^2) + a4 * exp(-((t-b4)/c4)^2) \quad (4.8)$$

The Coefficients:

a1 = 53.3

b1 = 1.522

c1 = 1.159

a2 = 461.3

b2 = 3.335

c2 = 1.052

a3 = 129.6

b3 = 2.261

c3 = 0.4826

a4 = 69.42

b4 = 4.372

c4 = 0.2981

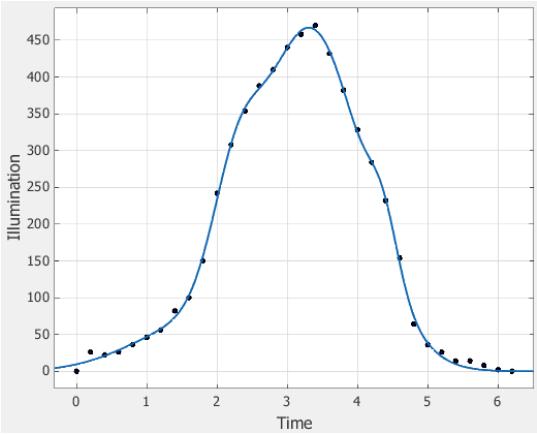


Figure 5.9 Plot of the actual data with 4th Gaussian derivative

The graph of the 4th Gaussian Derivative is illustrated in figure 5.8. A close observation of figure 5.8 reveals that only a few points of the actual illumination

values were not correctly predicted. Most predictions were close to the actual values giving a very high accuracy value of 0.999.

5.3.2.5 Smoothing Spline

The fifth mathematical equation investigated was the Smoothing Spline. It gives the best prediction accuracy results amongst all the mathematical equations experimented above. Thus theoretically, it is the best mathematical model to be used. Unfortunately, computationally, implementing a smoothing spline is difficult and would not be feasible in particular in computationally constrained environments such as implementing within a mobile device (See section 3.2.6).

Smoothing spline:

f(x) = piecewise polynomial computed from p

Smoothing parameter:

p = 0.99999401

Goodness of fit:

SSE: 0.1613

R-square: 1

Adjusted R-square: 1

RMSE: 0.7212

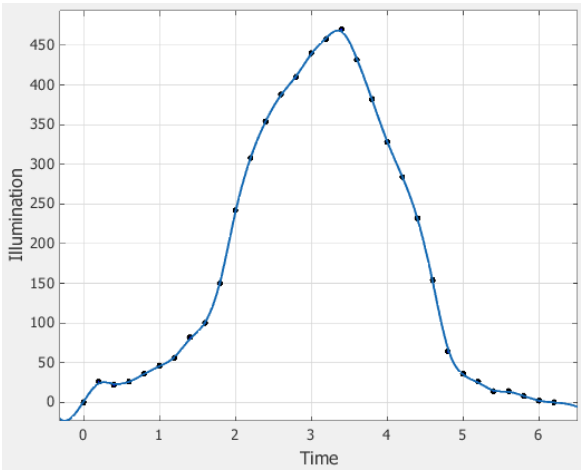


Figure 5.10 Plot of the actual data compared with a Smoothing Spline approximation

Figure 5.9 illustrates a perfect prediction where all the actual illumination values are correctly predicted by smoothing spline. However, due its complexity in

terms of implementation, it has been discarded for consideration towards practical implementation.

5.3.2.6 Sum-of-Sine

The use of a simple Sine function in the form of $I\ (t)\ =\ a * sin\ (b * t) + c$ was investigated in section 5.3.2.1 and was proven to give a high accuracy value of 0.85. It was also mentioned that implementing a Sine function of this nature is computationally very efficient and hence this is a feasible solution for implementing with a mobile device. Targeting a higher accuracy, in this section we investigate the use of Sum-of-Sine.

Table 5.5 Goodness of fit of 2nd and 4th Sum-of-Sine

Goodness of fit/ Sum-of-Sine	SSE	R-square	Adjusted R-square	RMSE
2nd	6017	0.9933	0.992	15.21
4th	3674	0.9959	0.9937	13.55

Table 4.5 illustrates the R-Square values of 2nd and 4th Sum-of-Sine. The accuracy of 4th Sum-of-Sine gives a higher value compared to 2nd Sum-of-Sine. However, the difference is insignificant between both R-Square values.

General model for 4th Sum-of-Sine:

$$I(t)\ =\ a1 * sin(b1 * t + c1)\ +\ a2 * sin(b2 * t + c2)\ +\ a3 * sin(b3 * t + c3)\ +\ a4 * sin(b4 * t + c4) \qquad (4.9)$$

Coefficients:

a1 = 311.3

b1 = 0.4863

c1 = 0.0582

a2 = 20.46

b2 = 2.434

c2 = -0.1797

a3 = 136.4

b3 = 1.466

c3 = -3.109

a4 = 15.08

b4 = 3.588

c4 = -0.4017

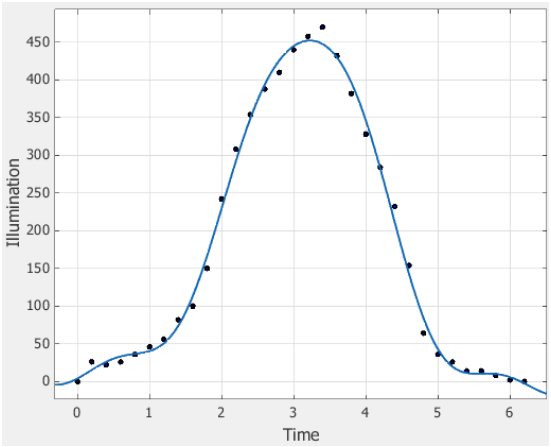


Figure 5.11 Plot of the actual illumination values and prediction based on 4th Sum-of-Sine

Figure 5.10 shows plots of the actual illumination values and the predicted ones using the equation of 4th Sum-of-Sine. It has given a high accuracy rate of 0.99 yet has only missed predicting one of the peak points.

5.3.3 Summary of Research Findings

After investigating the use of six different mathematical equations/models for predicting illumination variation that results from a single point light source (see section 5.3.2), finding the most accurate prediction using each mathematical model was conducted. The results thus obtained are presented in table 4.6.

Table 5.6 Comparison of the best result of each of the six different Mathematical models investigated

Goodness of fit/ Model	SSE	R-square	Adjusted R-square	RMSE
$f(x) = a.sin(bx - t)3$	7416	0.9893	0.9884	17.96
9th Degree Polynomial	4943	0.9945	0.9922	14.99
8th Fourier Harmonic	548.9	0.9994	0.9986	6.262
4th Gaussian Derivative	923.3	0.999	0.999	6.795
Smoothing Spline	0.1613	1	1	0.7212
4th Sum-of-Sine	3674	0.9959	0.9937	13.55

A close inspection of the results tabulated in table 5.6 reveals that theoretically the most accurate results depicted in the table is demonstrated by the use of the Smoothing Spline that gives the highest accuracy figures based on SSE, R-square, Adjusted R-square and RMSE. The prediction results obtainable by the use of the 8th Fourier harmonic and 4th Gaussian derivative are similar based R-square values measured. But pointing out that the best result is of the 8th Fourier harmonic. The 4th Sum-of-Sine function's goodness of fit gives it the fourth place along with 9th Polynomial degree showing very close results. The least accurate model option is the use of $I\ (t)\ =\ a.sin\ (bt - y)^3$ demonstrated by the least values of goodness of fit. Yet for this model the R-Square value obtained is a high 0.98 and can thus still be recommended for practical usage.

Table 5.7 The Mathematical Equations of the models presented in table 5.6

Model	Mathematical Equation
9th Degree Polynomial	$I(t) = p1 * t^9 + p2 * t^8 + p3 * t^7 + p4 * t^6 + p5 * t^5 + p6 * t^4 + p7 * t^3 + p8 * t^2 + p9 * t + p10$
8th Fourier Harmonic	$I(t) = a0 + a1 * cos(t * w) + b1 * sin(t * w) + a2 * cos(2 * t * w) + b2 * sin(2 * t * w) + a3 * cos(3 * t * w) + b3 * sin(3 * t * w) + a4 * cos(4 * t * w) + b4 * sin(4 * t * w) + a5 * cos(5 * t * w) + b5 * sin(5 * t * w) + a6 * cos(6 * t * w) + b6 * sin(6 * t * w) + a7 * cos(7 * t * w) + b7 * sin(7 * t * w) + a8 * cos(8 * t * w) + b8 * sin(8 * t * w)$
4th Gaussian Derivative	$I(t) = a1 * exp(-((t - b1)/c1)^2) + a2 * exp(-((t - b2)/c2)^2) + a3 * exp(-((t - b3)\ /c3)\text{^}2) + a4 * exp(-((t - b4)/c4)\text{^}2)$
Smoothing Spline	Multiple equations for each piece of the curve
4th Sum-of-Sine	$I(t) = a1 * sin(b1 * t + c1) + a2 * sin(b2 * t + c2) + a3 * sin(b3 * t + c3) + 4 * sin(b4 * t + c4)$
Sine Function	$I(t) = a.sin(b * t - y)^3$

The mathematical equations of all of the best performing models summarized above are tabulated in table 5.7. It shows that the implementation on handheld devices would be looked at from a different angle where the R-Square values should be sufficiently high but at the same time the equation used for the mathematical models should be computationally simple to implement. For the more mathematically complicated and demanding models, it is questionable if the hardware in a computationally constrained environment is capable to adapt to the real-time need of evaluating each predicted illumination value. It is noted that when an increased number of predictions/computations are required, the cost of the hardware will significantly increase.

A further table of results was prepared (see Table 5.8) based on the mathematical/ computational simplicity of equations and reasonably good prediction results. In particular the Smoothing spline was excluded from the comparison as it has a large number of equations and implementation wise it will be costly.

Table 5.8 Comparing the Most Practical & Efficient Models

Mathematical Model	R-Square	Equation
4th Polynomial degree	0.9554	$I(t) = p1 * t^4 + p2 * t^3 + p3 * t^2 + p4 * t + p5$
1st Gaussian Derivative	0.9882	$I(t) = a1 * exp(-((t - b1)/c1)^2)$
1st Fourier Harmonic	0.9789	$I(t) = a0 + a1 * cos(t * w) + b1 * sin(t * w)$
2nd Sum-of-Sine	0.9933	$I(t) = a1 * sin(b1 * t + c1) + a2 * sin(b2 * t + c2)$
Sine Function	0.8552	$I(t) = a * sin(b * t) + c$

Based on table 5.8, the Sum-of-Sine function seems to be the best choice. It results in the best accuracy of prediction (0.9933) for the most reasonable level of complexity of implementing two separate Sine functions.

5.4 Modelling Illumination in an Environment Lit Up by Two Point Light Sources

Typically handheld devices are used outdoors under natural lighting conditions or indoors under one, two or more artificial light sources. In all cases, the ALS will capture illumination values when the handheld device is on and is functioning.

In section 5.3, we investigated the use of six different mathematical functions to model and predict illumination distribution in the environment of a single point light source. Having rigorously investigated illumination distribution within the simplest possible lighting environment, this section takes the experiments a step further towards a real life scenarios by investigating the behaviour of the ALS of the handheld device under two light sources.

5.4.1 Experiments

The experiments were conducted indoors in a dark room where all the main light sources were turned off and external lighting was appropriately blocked out. A wooden surfaced table of 97.8 cm in length, 67.3 cm in width and 65 cm in height was kept at the centre of the room. Two LED light sources were hung on a tripod at 50 cm apart and 75 cm height above the surface of the table. The two light sources were in a horizontal line. The phone and the light were in the same vertical plane. The Nexus 7 was pulled at approximately constant speed from one end of the to the other end in a straight line to enable capturing illumination values.

5.4.2 Results and Analysis

The same six chosen Mathematical Models that were investigated for the one light source experiment were also applied in here as well. The results have over all reduced in terms of R-Square values as compared to those obtained with one light source experiment. This is unavoidable due to the added complexity of the lighting model. The sections 5.4.2.1-6 illustrate the results of the six models.

5.4.2.1 The Sine Function

In this experiment, the use of the Sine function, $I\ (t)\ =\ 916 * sin(0.2 * t - y)^n$ was investigated. Note that the values of 'a' and 'b' were determined empirically. It was found that the overall prediction accuracy results in the case of modelling illumination in a two light source environment is generally less accurate than in a single light source environment. However, the prediction accuracies obtainable are still low given that the highest R-Square value obtainable is 0.7785, for the 4th exponent.

Table 5.9 Goodness of fit obtainable with $I\ (t)\ = 916 * sin(0.2 * t - y)^n$

Exponents (n)/ Goodness of fit	2	4	6	8
R-square	0.5647	0.7785	0.7424	0.6641
Adjusted R-square	0.5647	0.7785	0.7424	0.6641
RMSE	192	137	147.7	168.7

The values in $I\ (t)\ = 916 * sin(0.2 * t - y)^n$ were chosen to fit this model. 916 represents the amplitude, 0.2 represents the period and t represents the phase shift. The odd exponents have given negative R-Square values due to the possibility of poor fit. It was noted that odd exponents result in negative R-Square values.

5.4.2.2 Polynomial Function

The second model investigated was the use of Polynomials of 4th, 5th and 6th degree. The results are tabulated in Table 5.11. The 4th degree polynomial demonstrated the highest prediction accuracy whilst the 5th and 6th degree polynomials failed to predict at a reasonable level of accuracy.

Table 5.10 Goodness of fit obtained with 4th, 5th and 6th degree of Polynomial

Goodness of fit/ Model	SSE	R-Square	Adjusted R-Square	RMSE
4th Degree Polynomial	3.122e+06	0.7711	0.7622	141.9
5th Degree Polynomial	7.096e+06	0.4798	0.4631	213.3
6th Degree Polynomial	8.392e+06	0.3847	0.3691	231.2

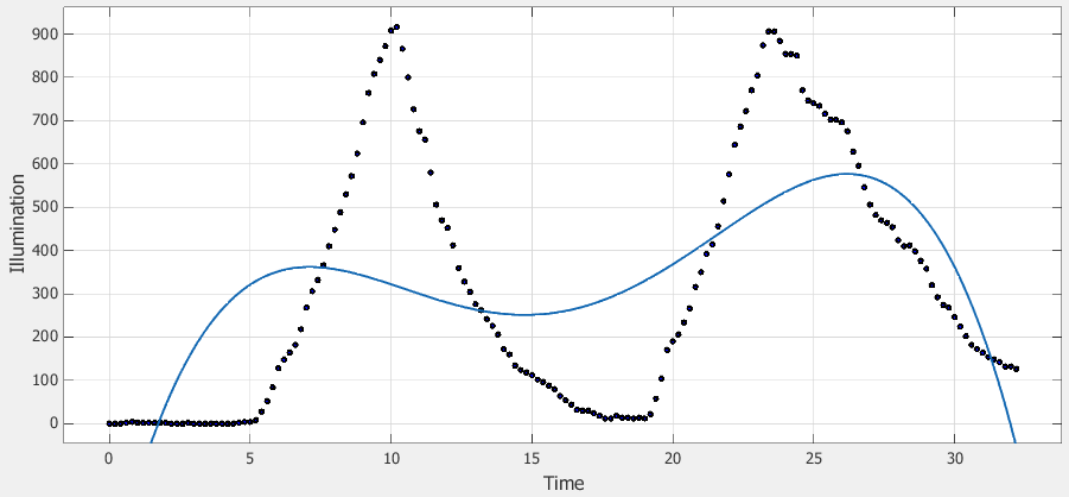


Figure 5.12 Predicted values based on a 4th degree polynomial vs. measured illumination values in the environment of two light sources

The highest R-Square value observed in table 5.11 for a Polynomial of degree four is still low. The graph of the 4th polynomial degree illustrated in figure 5.11 shows that the predicted values are not a good fit to the measured illumination values.

5.4.2.3 Fourier Harmonics

Prediction using the Fourier Harmonics has given very good level of accuracy. The lowest R-Square value obtained was 0.8569 for the 1st Fourier Harmonic. This value increased to 0.993 when the 8th Fourier Harmonic was used for predicting. Table 5.12 tabulates all experimental results obtained under this experiment. As the harmonic increases, the value of the R-Square noticeably increases.

Table 5.11 Goodness of fit obtained when using the Fourier Harmonics for modelling illumination in a two light source environment

Goodness of fit/ Model	SSE	R-Square	Adjusted R-Square	RMSE
1st Fourier Harmonic	1.952e+06	0.8569	0.8542	111.2
2nd Fourier Harmonic	1.475e+06	0.8919	0.8884	97.23
3rd Fourier Harmonic	9.465e+05	0.9306	0.9275	78.4

4th Fourier Harmonic	6.022e+05	0.9559	0.9532	62.94
5th Fourier Harmonic	1.862e+05	0.9863	0.9853	35.23
6th Fourier Harmonic	1.655e+05	0.9879	0.9868	33.44
7th Fourier Harmonic	1.281e+05	0.9906	0.9896	29.62
8th Fourier Harmonic	9.575e+04	0.993	0.9922	25.79

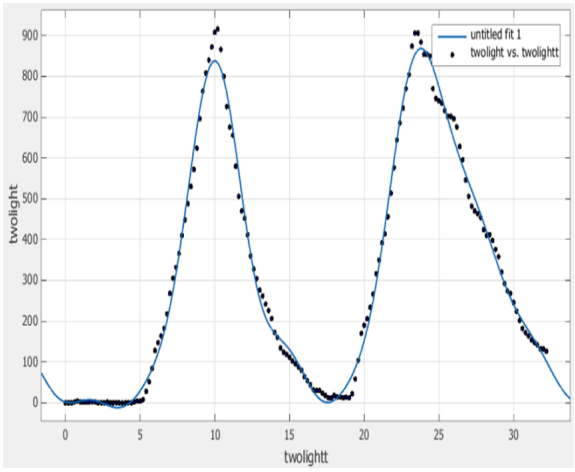


Figure 5.13 Use of 8th Fourier Harmonic for modelling illumination in a two light source environment

The graph of the 8th harmonic illustrated in figure 5.12 shows that the prediction based on the actual illumination values, is accurate. A closer look at figure 5.12 reveals some miss-predicted values at especially the peaks of the curve, yet the results are still more accurate compared to the previous models investigated.

5.4.2.4 Gaussian Derivative

In this section we investigate the use of Gaussian derivatives to predict illumination in a two light source environment. The Gaussian derivatives have given good prediction accuracies with the highest R-Square value obtained being 0.9974. Looking at the prediction accuracies of all the derivatives, none of them have resulted in a R-Square value below 0.9, indicating a very good accuracy of approximation.

Table 5.12 The goodness of fit obtainable with Gaussian Derivative

Goodness of fit/ Model	SSE	R-Square	Adjusted R-Square	RMSE
2nd Gaussian Derivative	5.53e+05	0.9595	0.9582	59.54
3rd Gaussian Derivative	1.911e+05	0.986	0.9853	35.34
4th Gaussian Derivative	1.609e+05	0.9882	0.9873	32.76
5th Gaussian Derivative	8.62e+04	0.9937	0.9931	24.22
6th Gaussian Derivative	3.908e+04	0.9971	0.9968	16.47
7th Gaussian Derivative	1.57e+05	0.9885	0.9869	33.37
8th Gaussian Derivative	3.596e+04	0.9974	0.9969	16.14

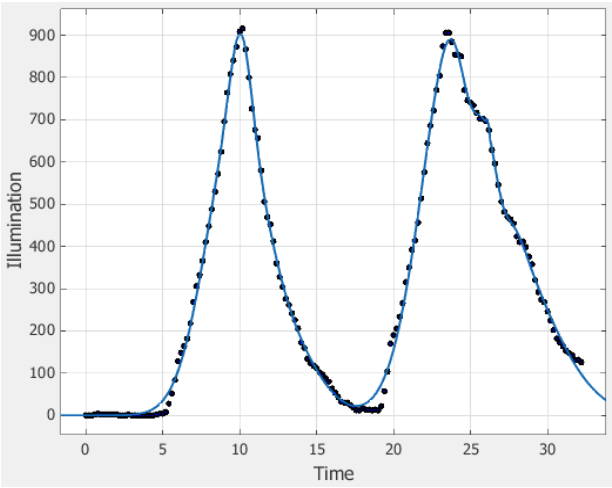


Figure 5.14 The use of 8th Gaussian Derivative to predict illumination in an environment of two light sources

The plot in figure 5.14 of the 8th Gaussian derivate illustrates a better prediction than the 8th Fourier harmonic.

5.4.2.5 Smoothing Spline

Similar to the accuracy demonstrated in the prediction of illumination in the environment of a single light source the Smoothing Spline demonstrates an excellent level of accuracy in prediction of illumination in the environment of two light sources.

Table 5.13 Goodness of fit values obtainable using the Smoothing Spline

Goodness of fit/ Model	SSE	R-Square	Adjusted R-Square	RMSE
Smoothing Spline	0.7648	1	1	0.6761

5.4.2.6 Sum-of-Sine

The 2nd and 3rd Sum-of-Sine produced the best accuracy of prediction with R-square values in excess of 0.9. Higher or lower order sums produced significantly lower prediction accuracies.

Table 5.14 Goodness of fit values for Sum-of -Sine

Goodness of fit/ Model	SSE	R-Square	Adjusted R-Square	RMSE
2nd Sum-of-Sine	1.154e+06	0.9154	0.9127	86.01
3rd Sum-of-Sine	9.327e+05	0.9316	0.928	78.08

5.4.3 Summary of Research Findings

The best R-Square values obtainable with the use of each of the six Mathematical Models investigated above have been summarized in table 5.16. The table indicates that theoretically the most accurate model is the Smoothing Spline that produced a R-Square value of 1. The 8th Gaussian Derivative produced the next best prediction accuracy closely followed by the 8th Fourier Harmonic. The 4th degree Polynomial produced the worst prediction accuracy giving a R-Square value of 0.77.

Table 5.15 The best Goodness of fit obtainable from all Mathematical Models for prediction of illumination lit up by two light sources

Goodness of fit/ Model	SSE	R-Square	Adjusted R-Square	RMSE
$I(t) = 916 * sin(0.2 * t - y)^5$	6.328e+07	0.7785	0.7785	137
4th Degree Polynomial	3.122e+06	0.7711	0.7622	141.9
8th Fourier Harmonic	9.575e+04	0.993	0.9922	25.79
8th Gaussian Derivative	3.596e+04	0.9974	0.9969	16.14
Smoothing Spline	0.7648	1	1	0.6761
3rd Sum-of-Sine	9.327e+05	0.9316	0.928	78.08

Based on the results tabulated in table 5.16, considering an implementation perspective, the best model to be used is the Sum-of-Sine. It gives an R-square value of 0.9316 for the 3rd Sum-of-Sine and the corresponding equation is computationally costly to implement as compared to the rest of the models that have higher R-Square values. Table 5.17 lists the prediction models in order of increasing complexity so that an informed decision can be made comparing accuracy vs. complexity.

Table 5.16 The best Mathematical Models to be used to model an environment with two light sources when viewed from an implementation perspective

Mathematical Model	Equation
Sine Function	$I(t) = 916 * sin(0.2 * t - y)^5$
4th Degree Polynomial	$I(t) = p1 * t^4 + p2 * t^3 + p3 * t^2 + p4 * t + p5$
8th Fourier Harmonic	$I(t) = a0 + a1 * cos(t * w) + b1 * sin(t * w) + a2 * cos(2 * t * w) + b2 * sin(2 * t * w) + a3 * cos(3 * t * w) + b3 * sin(3 * t * w) + a4 * cos(4 * t * w) + b4 * sin(4 * t * w) + a5 * cos(5 * t * w) + b5 * sin(5 * t * w) + a6 * cos(6 * t * w) + b6 * sin(6 * t * w) + a7 * cos(7 * t * w) + b7 * sin(7 * t * w) + a8 * cos(8 * t * w) + b8 * sin(8 * t * w)$
8th Gaussian Derivative	$I(t) = a1 * exp(-((t - b1)/c1)^2) + a2 * exp(-((t - b2)/c2)^2) + a3 * exp(-((t - b3)/c3)^2) + a4 * exp(-((t - b4)/c4)^2) + a5 * exp(-((t - b5)/c5)^2) + a6 * exp(-((t - b6)/c6)^2) + a7 * exp(-((t - b7)/c7)^2) + a8 * exp(-((t - b8)/c8)^2)$
3rd Sum-of-Sine	$I(t) = a1 * sin(b1 * t + c1) + a2 * sin(b2 * t + c2) + a3 * sin(b3 * t + c3)$

5.5 Illumination Modelling with Mathematical Models - Conclusions

The prediction accuracies obtainable in illumination modelling of environments lit up by single and two light sources using six different mathematical models presented in sections 5.3 and 5.4 above revealed that in terms of accuracy or prediction and simplicity of implementation, same mathematical models perform best in both cases. Table 5.19 summarizes the results.

Table 5.17 The Mathematical Models and the corresponding R-Square values obtained for environments lit up by one and two light sources

Model /R-Square	One light source	Two light source
Sine	0.9893	0.7785
Polynomial	0.9945	0.7711
Fourier	0.9994	0.993
Gaussian	0.999	0.9974
Smoothing Spline	1	1
Sum-of-Sine	0.9959	0.9316

A closer investigation of the results tabulated in table 4.19 reveals that the accuracy obtained in modelling illumination in an environment lit up by two light sources is generally lower than the accuracy obtained in modelling illumination in an environment lit up by a single light source environment. This is indicated by the fact that the R-Square value for two light sources has generally decreased compared to that of the one light source. However, the amount of decrease has varied from one model to another. The accuracy of the Sine function and Polynomial functions has noticeably decreased. In contrast the accuracy of the more computationally demanding mathematical functions such as the Fourier Series and the Gaussian derivatives have remained the same. In general, these two models are likely to work well modelling natural scenarios and trends, which may ensure that both of these functions would be able to model the illumination in a scene with multiple light sources. Further the Smoothing spline is likely to produce excellent prediction accuracies; however, its computational complexity will not allow it to be used practically. The Sum-of-

Sine has indicated a slight reduction of accuracy which leaves it a good candidate to be considered for modelling more complicated illumination distributions, given its low computational cost that makes it practically viable.

5.6 Effects of Distributed Multiple Light Sources

The experiments conducted above were focused on the modelling the illumination within an environment when one and two light sources were used respectively to light up the environment. Further the experiments were conducted under controlled conditions, i.e. no other light sources were present to illuminate the scene. There was no ambient light or light from diffused of unstructured light sources. The only possibility of changing the illumination distribution from a theoretical shape was the likely presence of secondary reflections, shadows and delays in measuring the sensor readings etc. This is however far from practical reality. In order to study the illumination distribution in a more practical and realistic environment in this section we present the results of a further experiment.

An experiment was conducted using a hand held mobile device to capture ALS reading when the user walked in the corridor of a building lit up by a regular and structured lighting system, i.e. light sources were equally and symmetrically spaced with naturel ambient light coming through windows opened to an outdoor space. At all times the user was holding the phone in a reading position. The graph in figure 5.14 illustrates the ALS reading changes. It is noted that the readings were taken when the user was on the move and hence any delay in the ALS's response would have an impact on the recorded readings. A close investigation of the graph reveals peaks reaching 1000 lux during the first 20 seconds and subsequently after 30 seconds approximately. In between the 20^{th} and 30^{th} second the peaks reduced to approximately 600 lux. By comparing with the arrangements of the lighting in the corridor and user's path of walking it was found out that peaks occurred when the user was directly under a light source fixed on the ceiling. During the 20^{th} to 30^{th} seconds the user was walking along a bend in the corridor where the angles of the light sources went through a change

with respect to the phones orientation. This reduced the overall level of illumination captured by the device.

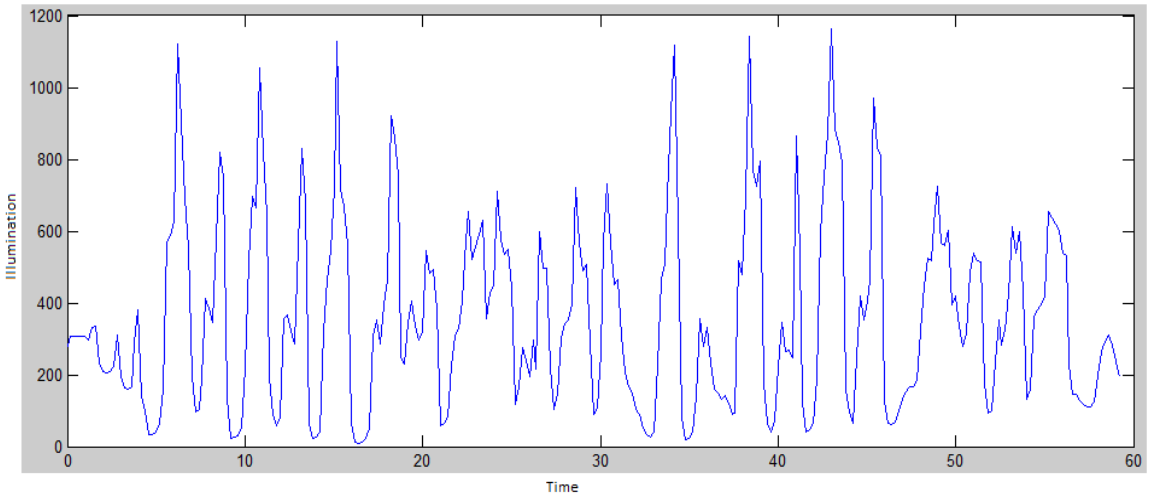


Figure 5.15 Illumination variations captured by the ALS of a mobile device carried by a user walking in a corridor lit by a uniform arrangement of light sources

If a person is standing in a corridor that is uniformly lit by symmetrically and equidistant spaced light sources, and is directly under one of the light sources fixed on the ceiling, the ALS is likely to pick illumination from not only the light source directly above, but also the light sources, closely around it. The angle at which the phone is held will result in which light source provides the peak response to contribute towards the ALS reading. More than one light source will contribute to each ALS reading typically as multiple light sources can be in the response range of the ALS. It is likely that the light sources that are placed at a significant distance from the phone will not contribute to the ALS readings, as they will be outside the response range of the ALS. Given that modern ALS sensors have a narrow angle of operation, the distance at which a light source in the ceiling may not have an impact of the overall ALS reading will not be that significant.

Whilst Gaussian, Fourier and Sum-of-Sine functions remain potentially good approximation function that can also be practically feasible for implementation both hardware and software wise, the complexities that arise from multiple light sources, reflections and discrepancies in the motion patterns of the user will

have to be considered in coming up with the best possible model. Under such circumstances and variations, the accuracy of the above models could significantly reduce. An alternative solution to using the mathematical models investigated in this chapter is to consider using Machine Learning algorithms. Supervised learning could be used to learn from known past data that can be captured during a device/App calibration routine. This aspect will be the subject of further research that will be presented inchapter-6.

5.7 Conclusion

In this chapter a rigorous investigation was carried out in using six different Mathematical Models for illumination modelling of an environment in which a mobile device is present. The mathematical functions investigated are; the Sine, Smoothing Spline, Gaussian Derivatives, Fourier Series, Sum-of-Sine and Polynomial functions. Based on the results obtained and the analysis of the results presented in this chapter, for one light source, theoretically the Smoothing Spline has given the best R-Square value of 1. On the other hand, practically, Sum-of-Sine Function seems to be the best option for practical implementation, given the reasonable R-Square value of 0.9933 and low computational cost.

For two light sources the results obtained are similar with the smoothing spline giving the best R-Square value of 1. On the other hand, practically, the Sum-of-Sine seems to be giving reasonably high accuracy at low computational cost.

Modelling the illumination in an environment lit by a single light source is considered to be a basic study of the illumination variation around the ALS of a mobile device. Modelling the illumination in an environment lit by two light sources is more challenging, but can yet be feasible using the six mathematical models investigated in detail in this chapter. However when the lighting becomes more complex such as in an environment lit up by multiple light sources the applicability of these mathematical models are questionable. In Chapter 6 we investigate the potential use of the mathematical functions in modelling the illumination of an area illuminated by light sources spaced in a 2D

set up. When the illumination becomes further complicated, such as in the case of considering diffused light sources, non-symmetric and/or random placement of light sources, the effect of ambient illumination, local/self-reflections, shadows etc., the mathematical models tested about could fail and alternative solutions may be required. In chapter 6 we investigate the use of machine learning algorithms to predict the illumination in an environment with complex lighting arrangements, i.e. those close to a real-life situation.

Chapter 6 : Illumination Modelling in an Environment of Multiple Light Sources

In the previous chapter, basic experiments were conducted to investigate the sensitivity and the operational performance of an Ambient Light Sensor (ALS) in the environment lit by either one or two light sources. In this chapter the work of Chapter-5 is further extended to investigate the modelling of illumination in an environment lit by multiple light sources. The light sources are placed in a horizontal, 2D plane in two lines with non-uniform spacing, whilst the mobile device that is fitted with the ALs is moved at a distanced 2D plane in various movement patterns, recording the ALS readings. It is noted that the lighting source distribution above is assumed to be close to an artificially lit indoor lighting set up and therefore of practical relevance. In order to model illumination we investigate the use of the computationally efficient mathematical models, Sum-of-Sine, experimented in Chapter-5. In addition to modelling illumination variations experienced by the ALS while it is being continuously moved, modelling of illumination based on static ALS readings obtained from the environment, using both polynomial functions and the use of simple machine learning algorithms is further investigated in chapter-6. The latter investigation is of practical relevance where one could assume that the modelling and estimation of illumination can be done based on a calibration stage where the mobile device picks up ALS readings obtained at few fixed points in the environment.

6.1 Introduction

In Chapter-5 the practical importance of the ability to model the illumination of an environment of a mobile device was explained. It was argued that such models could be used to overcome two common shortcomings of an ALS, namely the problem of narrow angle operation and the latency of operation. It was proven that simple mathematical models could be used to for accurately modelling the illumination in an environment lit up either by one or two light

sources. These models are likely to fail or be less accurate in predicting illumination in environment lit up by multiple light sources. Therefore it is important that experiments conducted in Chapter-5 are repeated and the suitability of each model is re-investigated in modelling illumination in such environments.

The modelling of illumination as measured by an ALS under their known operational constraints can be carried out considering two operational models of a typical mobile device's usage. Firstly, the ALS may attempt to promptly capture illumination values while it's being continuously moved and the models will require predicting illumination real-time, on the move. Secondly, it can be assumed that the user is requested to calibrate the mobile device in the environment by recording illumination, whilst being stationary momentarily at a number of different positions in the environment.

Given the above common operational scenarios of mobile handhelds, in section 6.2 we investigate the use of modelling illumination while on the move and in section 6.3 to model the illumination based on a small number of ALs readings recorded at a number of points in the environment. Environments with both regularly and non- regularly arranged lighting sources are investigated. Finally section 6.4 concludes with an insight into the research conducted in Chapter-7.

6.2 Illumination Modelling: Multiple Light Sources in a Structured Arrangement – Phone Moving

Experiments were carried out to model the illumination variation in an environment where the light sources were arranged in two horizontal parallel rows, spaced regularly and irregularly. Further the focus of the experiments presented in this section is to model the illumination changes that occur when the phone is being moved, rather than estimating the illumination at a stationary point in the environment. It is noted here that the practical relevance of the need to model the illumination as captured by a moving mobile handset was stated in Chapter-1.

6.2.1 Mathematical model

In chapter 5, six different regression models were implemented and compared based on the use of Polynomial, Sine, Gaussian, Sum-of-Sine, Fourier and Smoothing Spline, functions. It was concluded that the most cost-effective model that can be used for the modelling of illumination for one or two light sources is the Sum-of-Sine. Consequently in this chapter, the Sum-of-Sine was chosen as the mathematical model to be used.

6.2.2 Methodology

The goal of the proposed experiments is to enable the determination of an illumination model that will be able to predict illumination in an environment lit up by two lines of regularly and irregularly spaced light sources, set up in a horizontal planes. Satisfying the above requirement, three main experiments setups were designed which are considerably close to real life lighting infrastructure in buildings. However it is noted that some lighting configurations associated with the experiments studied in detail in this chapter rarely occurs in real life but was yet investigated to cover most possible cases.

Once the experiments are completed the ALS values are captured and subsequently transferred to a computer for detailed analysis. The mathematical function, Sum-of-Sine was used for modelling all captured illumination distributions due to its simplicity and efficiency as proven in Chapter-5.

The modelling of illumination was carried out for illumination values captured by the phone when the phone was moved in the environment in a horizontal plane in different paths as explained in the following section.

6.2.3 Experimental Setup

Samsung Galaxy S4 was used in recording the ALS readings in all experiments although high-end mobile phones of other makes could have been used. The reason for using this particular mobile phone was that it includes all the sensors needed for experiments conducted in this thesis, namely the ALS, Gyroscope and Accelerometer. However, in the experiments conducted in this chapter, the only

sensor that is required for data capture was the ALS. A dedicated mobile application was developed to capture the raw data from all the associated sensors. The data was then transferred to an external computer for computationally extensive processing / modelling.

Figure 6.1 illustrates the experiments conducted. The mobile device was initially placed in point x and was connected to a USB (data/charger) cable to enable pulling it at approximately constant speed, towards point y. The light sources (depicted as coloured circles) in figure 6.1, were fixed on four tables set as indicated. The tables provided a means to easily set up a measurement gauge to measure distances between light sources and to move the light sources smoothly and freely as required.

Three different setups of the light sources were chosen for this experiment (See Setup A, B and C described in sections 6.2.3.1, 6.2.3.2 and 6.2.3.3 respectively). In all three setups, three different scenarios of moving the phone from point x to point y were implemented as follows:

- Path-1: The phone was moved in a straight line starting from point x and heading towards point y (Figure 6.1: 1). This scenario was chosen because it is the most likely way for a user to move from one point to another.
- Path-2: The phone was moved in a slightly wavy path from point x heading towards point y (Figure 6.1: 2). This mimics a situation where a user is not particularly paying attention to walking in a straight path between two points.
- Path-3: The phone was moved in a significantly wavy line (compared to that of path-2), starting from point x and was heading towards point y (Figure 6.1: 3). This mimics a user behaviour that attempts forcefully to complicate the illumination modelling and prediction process.

In all above experiments the measurement of the ALS readings was originated at point X at which the application gathering sensor data readings was switched on.

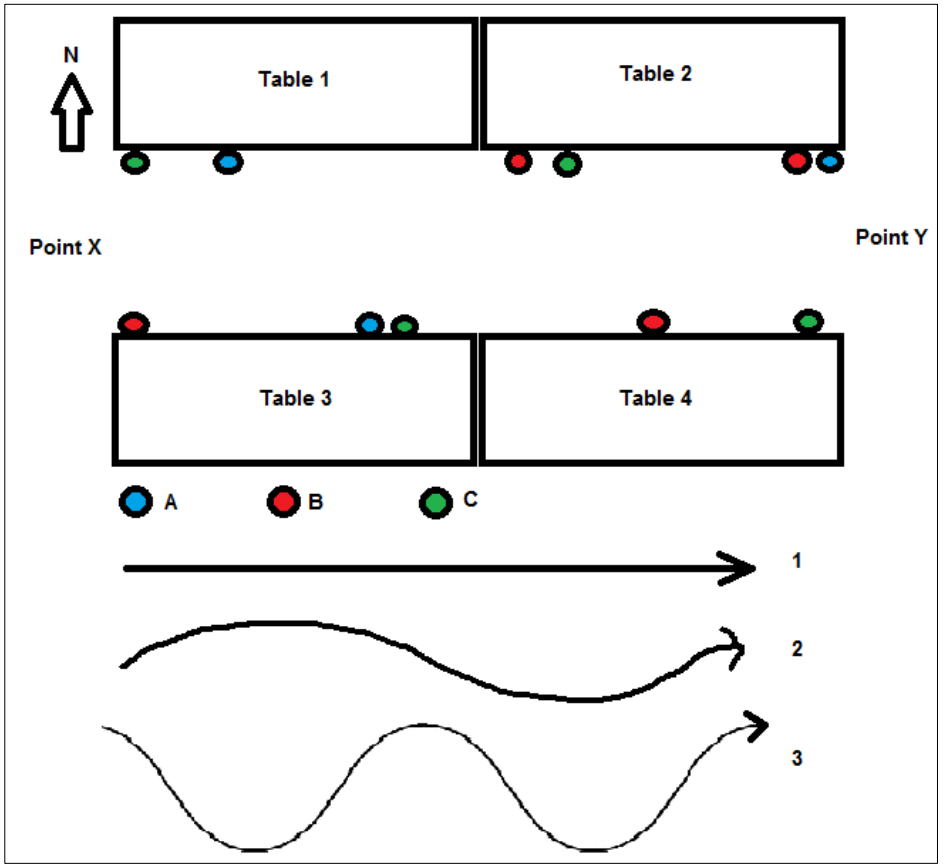


Figure 6.1 Set up for the experiments

As mentioned before three different lighting configurations were considered, namely, Setup A (blue circles in figure 6.1), Setup B (red circles in figure 6.1) and Setup C (green circles in figure 6.1). It is noted that Setup C represents a lighting configuration similar to that of a real life indoor lighting distribution where the light sources are placed on either side of the path placed at regular distances in a structured manner. It is noted that in each of the lighting configurations above, the lights sources that represent that setup are the only ones that are turned 'on' during that specific experiment. The normal room lighting was turned off and the room was converted to a dark room on a temporary basis.

Following sections describe the modelling of illumination under each of the different lighting configurations (i.e. setups) and device movement paths.

6.2.3.1 Setup A

The light sources indicated in blue colour in figure 6.1 are the only light sources turned on in this lighting configuration. The experiment was conducted in an improvised dark room facility. It is noted that the three light sources are placed in an irregular pattern.

6.2.3.1.1 Scenario 1

The phone was initially placed at point x and then pulled at approximately constant speed, in a straight line, towards point y. The data recording was initiated at point X and ALS readings were taken at regular time intervals until the phone reached point Y.

The Sum-of-Sine function as defined by equation (6.1) below was used to model the distribution of the variation of illumination with time.

$$I(t) = a1 * sin(b1 * t + c1) + a2 * sin(b2 * t + c2) \quad (6.1)$$

In figure 6.2 below, the black dots illustrate the measured illumination values and the blue continuous line illustrates the Sum-of-Sine model that best fits with the measured data. It is noted that the Y-axis indicates the actual illumination values while the X-Axis indicates the time.

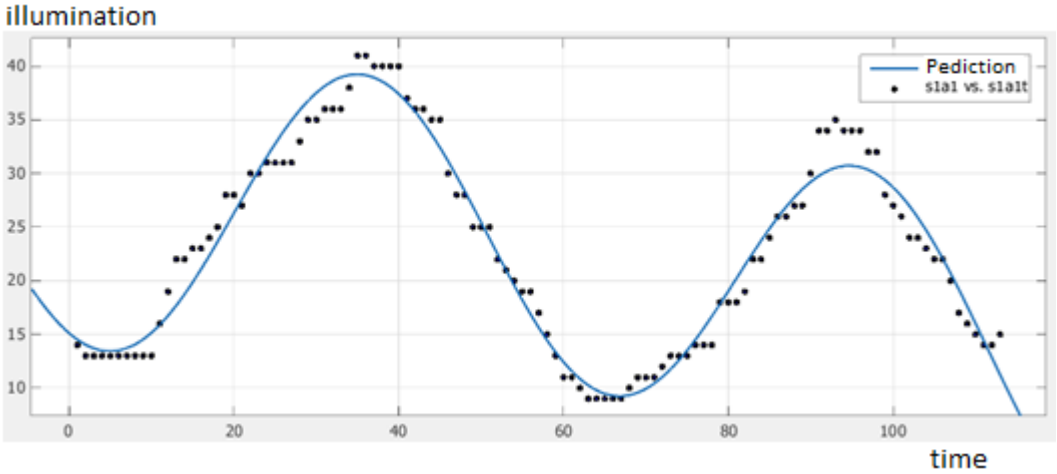


Figure 6.2 Model for Scenario 1 illumination variation

The Sum-of-Sine model fitting resulted in a R-square value of 0.9615, demonstrating an excellent overall level of fitness of the model to the measured values.

A close inspection of figure 6.2 reveals that in the first peak that occurs between 85 and 100 seconds, few points were missed out being accurately modelled. However, 35 and 40 between seconds, it was able to more accurately predict the peak points. A closer inspection of figure 6.2 reveals that the first peak occurred when the phone reached the centre point between the two light sources that were placed on tables, 1 and 3. This resulted in having a higher peak equal to 42 lux compared to the second peak that is 35 lux. The second peak is a result of one light source, i.e. the phone getting closer to the light source kept on table 2.

6.2.3.1.2 Scenario 2

The phone was moved in a slightly wavy line from point x, towards point y at an approximately constant speed. A Sum-of-Sine function with three terms was used in modelling the illumination as described in equation (6.2)

$$I(t) = a1 * \sin(b1 * t + c1) + a2 * \sin(b2 * t + c2) + a3 * sin(b3 * t + c3) \quad (6.2)$$

Applying a Sum-of-Sine function with two terms gave a low R-Square value of 0.4248. Applying a Sum-of-Sine function with three terms resulted in a higher R-square value of 0.746. However this accuracy was still approximately 25.24% less than that obtained when the phone was moved in a straight line.

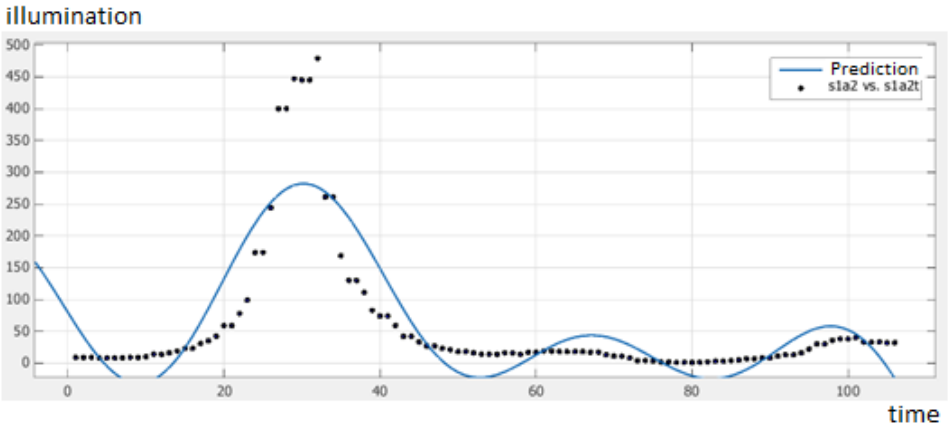


Figure 6.3 Model for Scenario 2 illumination variation

Figure 6.3 models the illumination variation of the Scenario-2. It shows that there is a single dominant peak occurring at approximately the 30th second. As described before this point corresponds to the time when the phone is placed in between the two light sources kept on tables 1 and 3 (see figure 6.4). A closer investigation of the model also reveals that the Sum-of-Sine model failed to predict the illumination values that are higher than 300 lux near the above peak. A further non-dominant peak exists closer to the 100-second point and this point corresponds to the case when the phone has reached the proximity of the light source kept on table 2. As the light was heading closer to the final light source placed in table 2, the illumination value illustrates a steady increment.

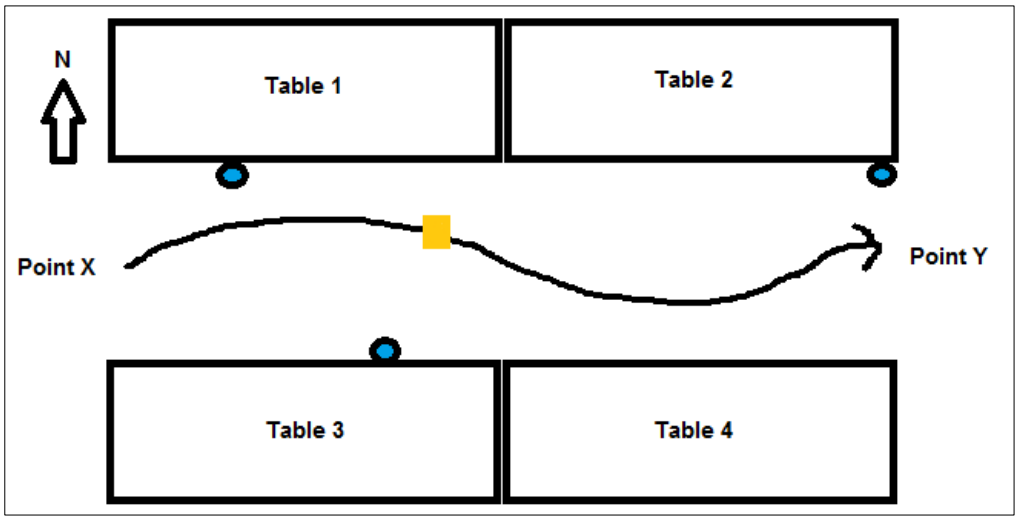


Figure 6.4 Setup A, Scenario 2

In Scenario 2 the light sources are not regularly and symmetrically placed making the illumination distribution more complex. It has so far being revealed that with scenario 2 a Sum-of-Sine function with three terms are needed to achieve any reasonable accuracy of prediction as against only needing a Sum-of-Sine function with two terms in the case when the phone was moved in a straight line. The added complexity of illumination caused both by the non-regular placement of lighting sources and the complex movement path of the mobile device, requires a more complex function to model the illumination variation.

6.2.3.1.3 Scenario 3

In this case the phone is moved in a more complex pattern (i.e. more wavy manner) as compared to the movement in scenario 2. In order to model the illumination to obtain a reasonable level of accuracy a Sum-of-Sine model having at least 4 terms had to be used as shown in equation 6.3.

$$I(t) = a1 * \sin(b1 * t + c1) + a2 * \sin(b2 * t + c2) + a3 * sin(b3 * t + c3) + a4 * sin(b4 * t + c4) \quad (6.3)$$

The model results in a R-square value of 0.7566 that is approximately 1.41% higher than when the phone was moved in a less wavy manner in scenario 2 where a Sum-of-Sine function with three terms was used.

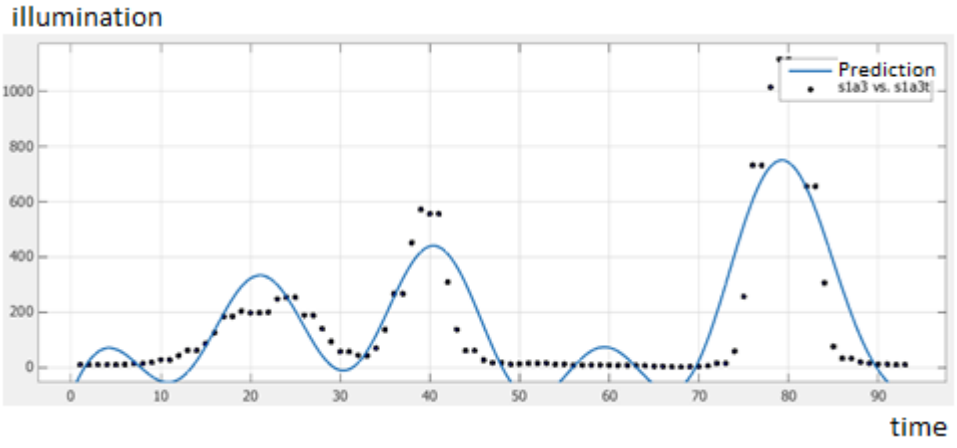


Figure 6.5 Model for Scenario 3 illumination variation

Due to the rather complex movement pattern that results in the phone getting closer to the individual light sources and passing points that are illuminated by more than one light source (see figure 6.6), more peaks of illumination values are indicated in the plot of illumination depicted in figure 6.5. This is also the reason for needing more sine terms in the function to model the illumination.

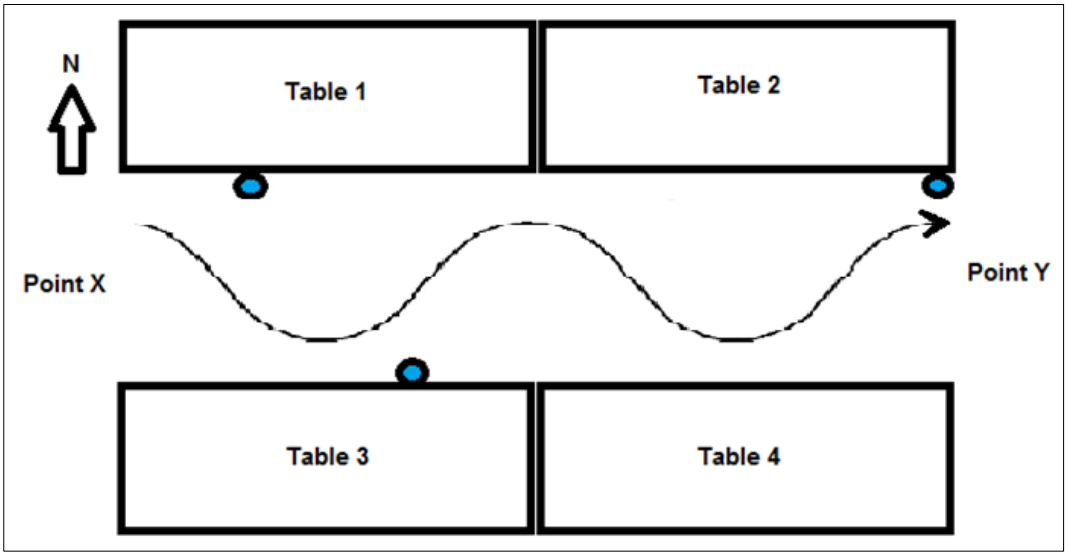


Figure 6.6 Setup A scenario 3

6.2.3.2 Setup B

The light sources in red colour are the only ones turned on in this setup (see figure 6.7). One light source was placed at the far end of table 3; two were placed in each corner of table 2 and one exactly in the centre of table 4. The area opposite and in between table 2 and 4 produces the highest level of illumination due to being illuminated by three light sources.

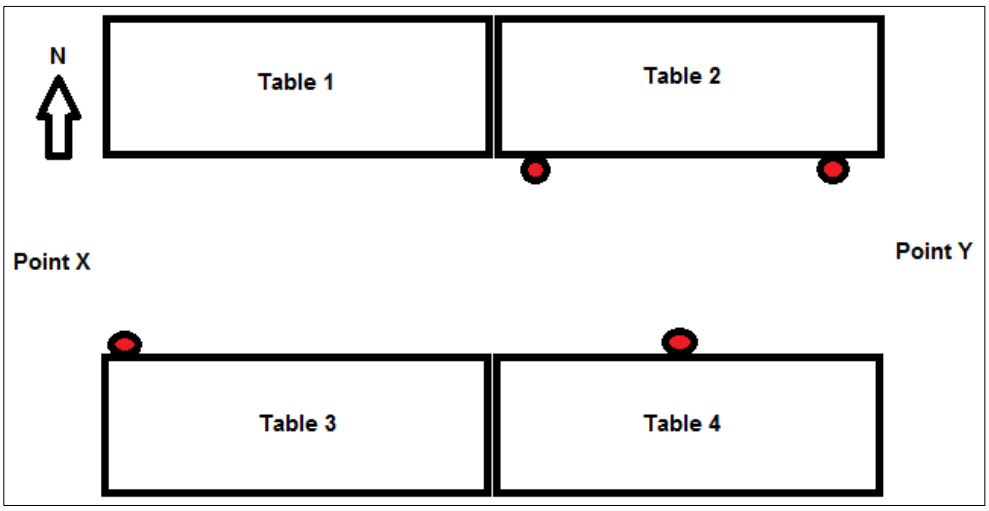


Figure 6.7 Light source configuration of Setup B

6.2.3.2.1 Scenario 1

In this case the phone was moved in a straight line from point X towards point Y. The following Sum-of-Sine function was used to model the illuminations variation (see figure 6.8).

$$I(t) = a1 * sin(b1 * t + c1) + a2 * sin(b2 * t + c2) \quad (6.4)$$

The model above produced an R-square value of 0.9198.

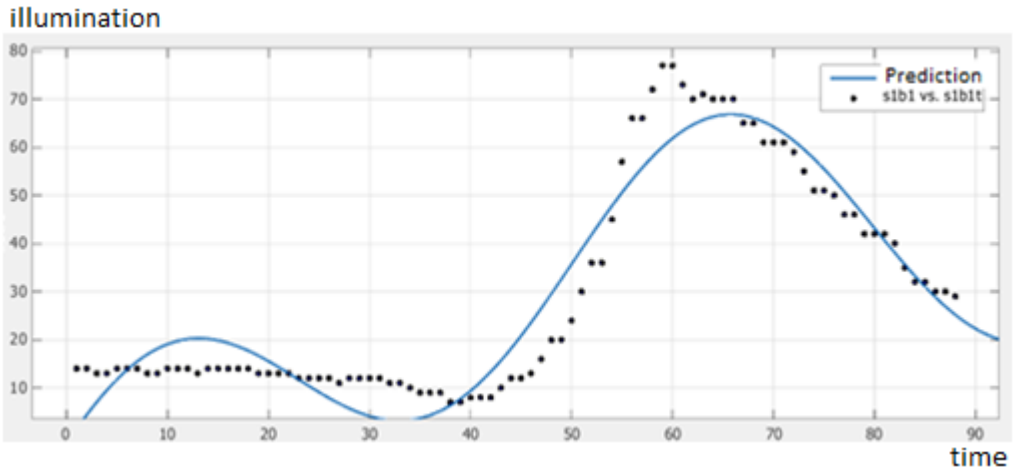


Figure 6.8 Implementation of Sum-of-Sine term 2

From the start, heading towards the 40th second, the illumination value reduced from 14 lux to 5 lux. The darkest point was reached around the 40th second and is the point that receives minimum amount of light from all four light sources.

After this point in time, a steady rate of increment commenced and the illumination value reached its peak around 60 seconds. At this point the phone is kept approximately at the centre of the triangle that is formed by the locations of light sources placed on the table 2 and table 4. After the 60th second, it is noticed that the rate of decrease in the illumination value is not sharp. The phone is moved in an area strongly illuminated by three light sources and hence the rate of decrease will be low.

6.2.3.2.2 Scenario 2

In this case the phone is moved in a slightly wavy pattern (from a straight line) from point X, towards point Y. To model the illumination a Sum-of-Sine function with three Sine terms, as described in equation 6.5, was used.

$$I(t) = a1 * \sin(b1 * t + c1) + a2 * \sin(b2 * t + c2) + a3 * sin(b3 * t + c3) \quad (6.5)$$

The model produced an R-square value of 0.8469, a figure that is approximately 8.25% less than that obtained for the model predicting illumination variation under scenario 1. Note that even with more Sine terms the accuracy obtainable is lower due to the complex pattern of illumination change.

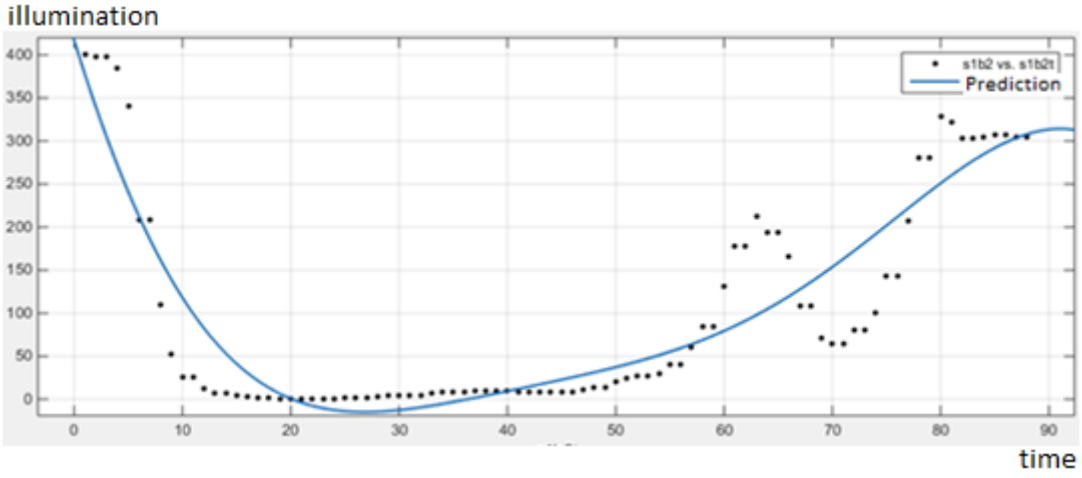


Figure 6.9 Illumination variation for Scenario-2, Setup B and prediction

Figure 6.9 illustrates the measured illumination variations and the predicted illumination obtained by the Sum-of-Sine function of equation 6.5. A closer examination of the graph, considering the wavy pattern of movement shows that peaks result when the phone gets closer to the four light sources, intern.

6.2.3.2.3 Scenario 3

In this case the phone was moved in a more wavy path as compared to the patch of scenario 2 starting from point x and ending at point y. A Sum-of-Sine function of four Sine terms was used to model the illumination variation.

$$I(t) = a1 * \sin(b1 * t + c1) + a2 * \sin(b2 * t + c2) + a3 * sin(b3 * t + c3) + a4 * sin(b4 * t + c4) \quad (6.6)$$

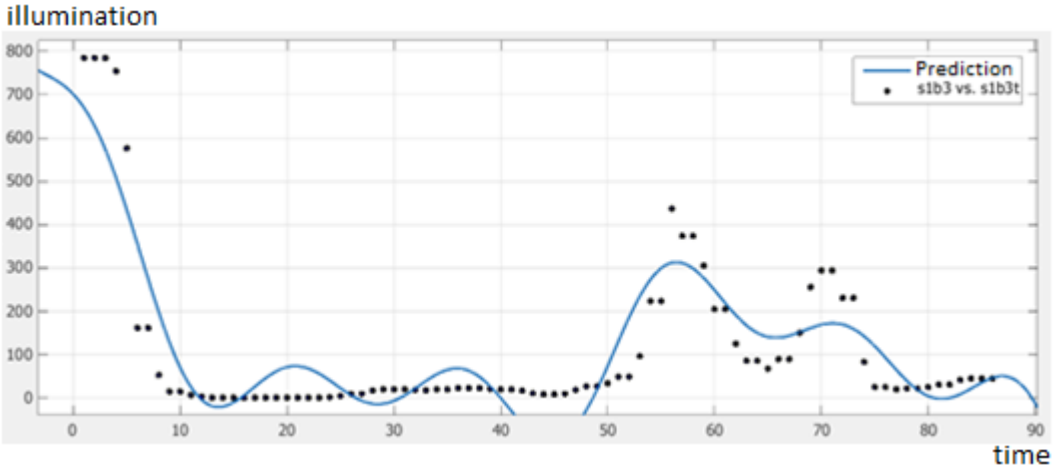


Figure 6.10 Illumination variation for Scenario-3, Setup B and prediction

The model produced a R-square value of 0.8205, which is approximately 8.10% less than when the phone was moved in a less wavy line. The variation of illumination is more complex than in the case of the scenario 2 and even with a more complex model, the accuracy obtainable has reduced. However for most practical purposes this is reasonable prediction accuracy.

6.2.3.3 Setup C

In this scenario only the light sources depicted in green colour are used as part of the lighting configuration. Although complex, this is the type of lighting configuration is that what is usually found in indoor environments.

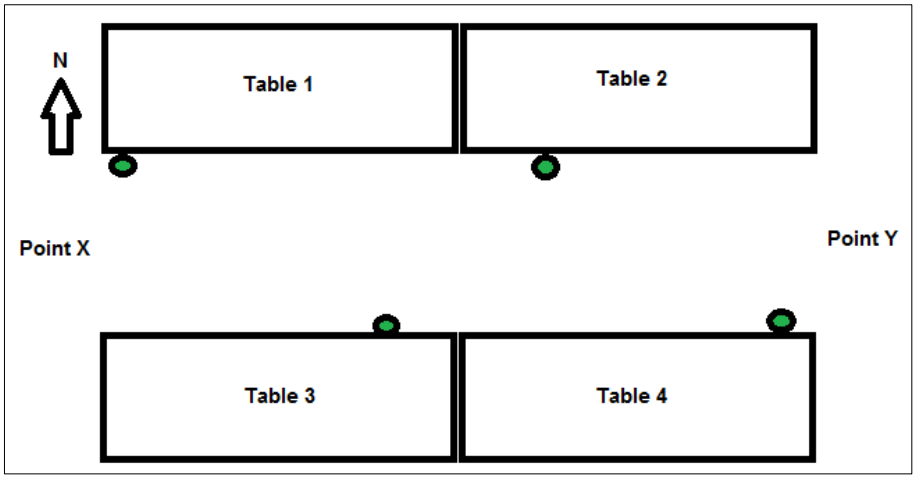


Figure 6.11 Light source configuration of Setup B

6.2.3.3.1 Scenario 1

In this scenario the phone is moved in a straight line starting from point X towards point Y. A Sum-of-Sine function with two Sine terms was used to model the illumination as defined in equation 6.7.

$$I(t) = a1 * sin(b1 * t + c1) + a2 * sin(b2 * t + c2) \quad (6.7)$$

The R-Square value obtained for the above model is 0.9308. Figure 6.12 plots the measured and predicted illumination values. A closer examination of the measured illumination values indicates the local maxima that happen when the phone moves between two light sources and is approximately placed in the mid point joining two light sources. In general the model used for prediction has been able to predict sufficiently accurately despite a significant prediction error closer to the 20th second.

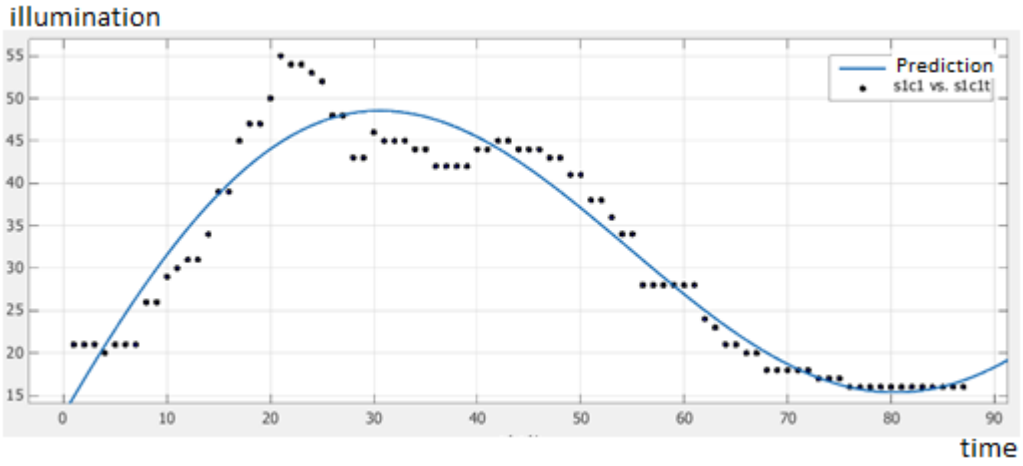


Figure 6.12 Illumination variation for Scenario-1, Setup C and prediction

6.2.3.3.2 Scenario 2

In this case the phone was moved in a slightly wavy path from point X towards point Y. A Sum-of-Sine function with four Sine terms (see equation 6.8) was used to model the illumination variation.

$$I(t) = a1 * \sin(b1 * t + c1) + a2 * \sin(b2 * t + c2) + a3 * sin(b3 * t + c3) + a4 * sin(b4 * t + c4) \quad (6.8)$$

The model produced an R-Square value of 0.8138. This is a considerable reduction of prediction accuracy as compared to that of the scenario-1. The increased complexity of the illumination variations caused by the movement of the mobile phone outside a straight-line path has resulted in this reduced prediction capability. The highest inaccuracies have been caused near illumination peaks.

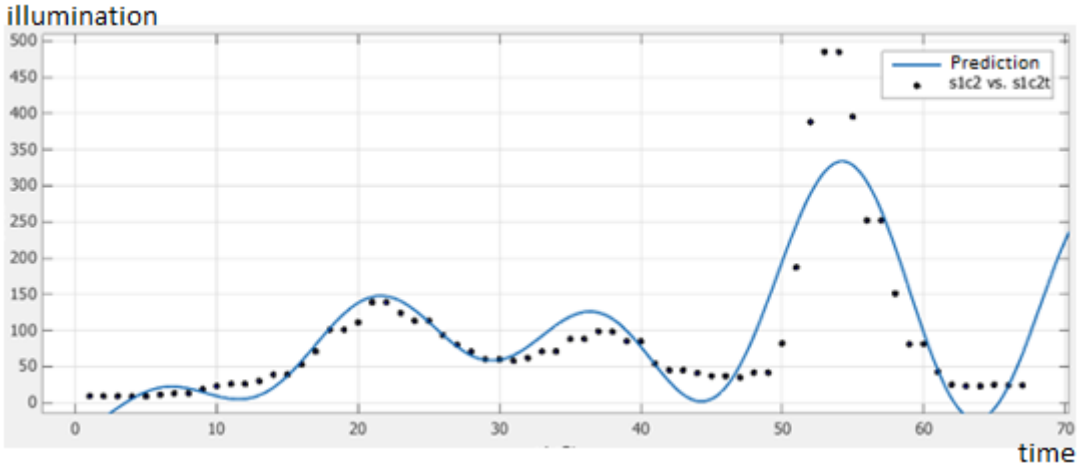


Figure 6.13 Illumination variation for Scenario-2, Setup C and prediction

6.2.3.3.3 Scenario 3

In this case the phone was moved in a path that is more complicated than that of the case of scenario-2. A Sum-of-Sine function with four Sine terms was used for modelling the illumination.

$$I(t) = a1 * \sin(b1 * t + c1) + a2 * \sin(b2 * t + c2) + a3 * sin(b3 * t + c3) + a4 * sin(b4 * t + c4) \quad (6.9)$$

The model only produced an R-square of 0.6007, which is significantly lower than that produced by the models used for all other scenarios and light source setups. The complicated lighting configuration adds complexity to the complex changes of illumination that has resulted from complex motion pattern of the mobile device.

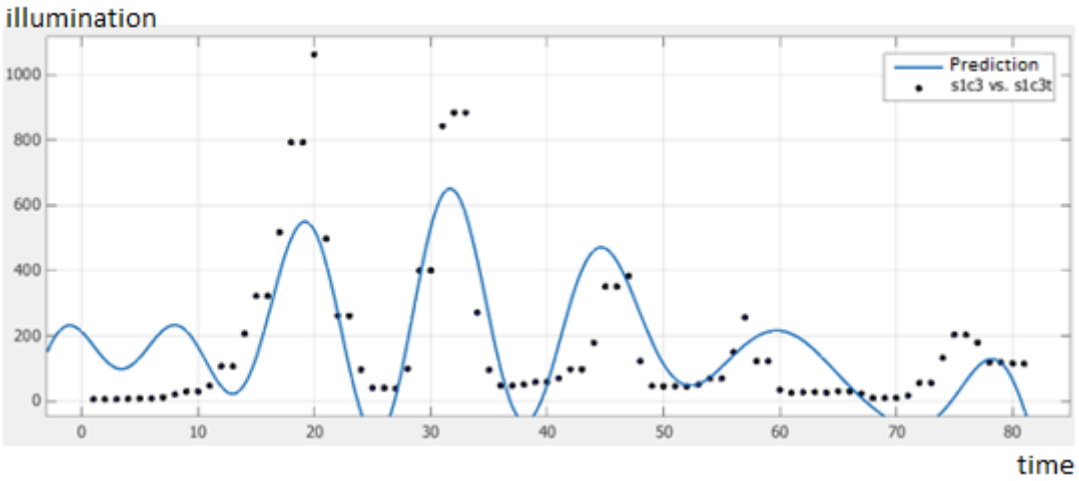


Figure 6.14 Illumination variation for Scenario-3, Setup C and prediction

6.2.4 Summary and Analysis of Prediction Results

In section 6.2 a number of experiments were conducted to measure the illumination surrounding a mobile device, as measured by the devices own Ambient Light Sensor. The ability to model such illumination variations using mathematical models, in particular the use of the Sum-of-Sine function, was investigated in detail. The intention of the experiments completed was to analyse the complexities that arise from considering increasingly complicated lighting configurations and mobile device movement patterns. Table 6.1 summarizes the R-square values obtained for each experiment conducted.

Table 6.1 R-Square value for all three setups and their three scenarios

		R-Square value of		
	Setup	A	B	C
Scenario	1 (straight line)	0.9615	0.9198	0.9308
	2 (wavy line)	0.746	0.8852	0.8138
	3 (wavier line)	0.7566	0.8205	0.6007

A close investigation of the prediction accuracies tabulated in table 6.1 reveals that in general when the complexity of the movement patterns and the complexity of the lighting configuration increases, the prediction accuracy decreases. This is easily observable from the fact that in the table 6.1 the best R-

Square value obtained was for setup A, scenario 1 (a value of 0.9615) and the worst obtained was for setup C, scenario 3 (a value of 0.6007).

In order to further investigate the above observation a further experiment was conducted for the cases, setup A, scenario 1 and setup C, Scenario 3 using Sum-of-Sine containing 1-4 Sine terms and using Fourier series of 1-4 order, inclusive. In general the tabulated results indicate that despite using more complicated higher order functions to model illumination variation, all mathematical models fail when predicting the illumination values in an environment illuminated by more complex lighting patterns and is supporting a free movement of a phone.

Table 6.2 Comparison between using Sum-of-Sine and Fourier Series of different orders in modelling illumination variations in Setup A, Scenario 1 and Setup C, Scenario 3

Setup A Scenario 1				Setup C Scenario 3			
Fourier Series		Sum-of-Sine		Fourier Series		Sum-of-Sine	
Harmonic	R-Square	Term	R-Square	Harmonic	R-Square	Term	R-Square
1st	0.7993	1	0.0308	1st	0.271	1	0.1194
2nd	0.9511	2	0.9615	2nd	0.3098	2	0.3493
3rd	0.9663	3	0.9668	3rd	0.3475	3	0.571
4th	0.9808	4	0.9802	4th	0.3857	4	0.6007

Observations made in the experiments presented in section 6.2 concludes that for more complicated lighting configurations depicting lit up indoor and/or outdoor environments, it will be required to use other mathematical models that promises more efficient and accurate modelling capabilities. To this effect in the rest of the chapter we investigate the use of different order polynomial functions and using machine learning algorithms, replacing the use of Sum-of-Sine and the Fourier Series in modelling illumination in potential practical situations, i.e. environments lit up by complicated distributions of light sources.

6.3 Illumination Modelling in Complex Lighting Distributions

Artificially lit up indoor areas usually consist of many individual light sources arranged mostly in regular patterns, i.e. for example ceiling light configurations in a large office environment. However due to additional light sources that could exist in such a space (e.g. additional light sources such as pedestal, desk/table lights, such regular lighting patterns can become irregular. The impact of natural light to such scenes can also add a further level of complication.

Given the above fact one should endeavour to model more illumination in environments of more complex and random arrangement of light sources. One further observation is that in most practical situations it may not be essential to model the illumination variations while the mobile user is moving. For most practical cases a typical user will focus on a mobile device screen while stationary, either sitting or standing. Therefore in the subsequent experiments conducted in this chapter we record the ALS readings at different positions in an environment while the user is stationary. The assumption is, the data required to train the models that can mimic the illumination variation within a space will be naturally captured by a user who may be freely moving within such a space, initially, but will be momentarily static (i.e. not moving) at least at few places within the environment. For example during natural random movement of a user, the device can capture stable illumination readings at several different locations that will provide the known data for creating the illumination models.

Given the above observations in this section we conduct new experiments in order to capture, model and predict the illumination of scenes that are illuminated by complex and random collections of light sources. Two experiments are conducted as follows:

6.3.1 Experiment 1 – Environment Lit by Symmetric Light Sources

In this experiment ALS readings were captured at a certain number of points in an environment that was lit by light sources placed regularly and symmetrically. In taking all the measurements the mobile device was held by the user, still and it's plane approximately horizontal.

Figure 6.27 illustrates the experimental setup. Points labelled x1, x2, x3 and x4 represent the positions where the mobile device was placed to capture the illumination values. The circles in figure 6.27, labelled 1,2,3 and 4, represent the locations of the light sources. The light sources are placed in a horizontal plane approximately 65 cm above the surface where the mobile phone was kept and moved. The light sources are identical and individually controllable to switch on/off as needed, providing the capability create a large number of lighting configurations from the four light sources.

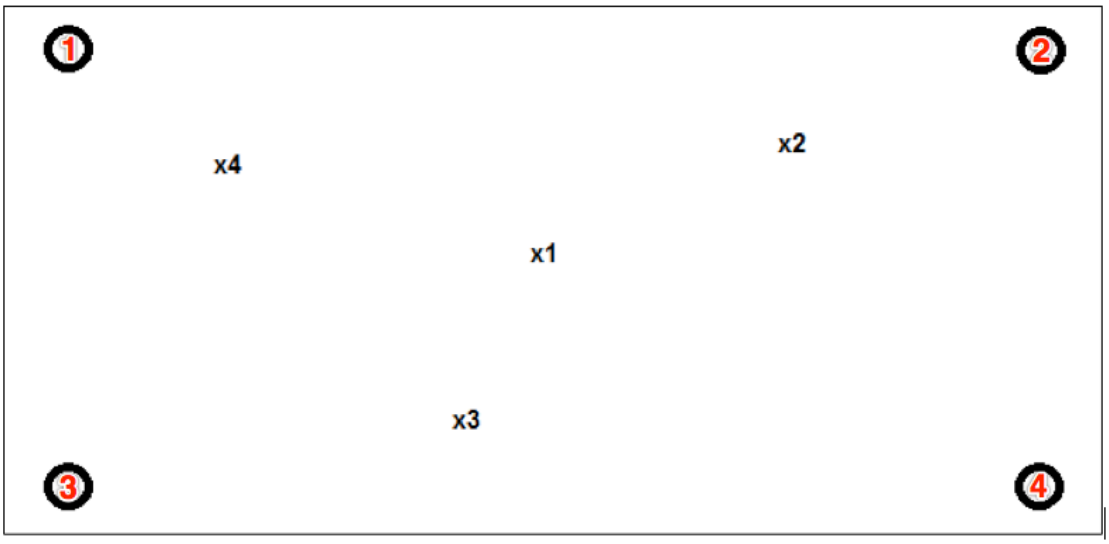


Figure 6.15 Setup of Experiment 1

Assuming that the exact position of each point at which the mobile device is kept can be measured with respect to the locations of the light sources, it is possible to use this information to define the illumination value at any point in the environment by generating an illumination model.

All light sources are identical producing an illumination value 1150 lux at very close proximity. This value reduces with distance from the light source. Directly under a light source the illumination reading is 1150 lux that is equal for all the light sources. The closer the phone will be from the light source the higher the lux value will be and the farther it is from the light source, the lower the illumination value will be.

The phone is placed in position x1, held still and the illumination value measured was found to be 39. Placing the phone in a fixed position to capture the illumination value results in stability of the rest of the sensors, namely accelerometer and gyroscope. There are minor fluctuations in sensor readings due to gravity but they are too insignificant to be considered. Once the illumination value is recorded the next measurement taken is the mobile phone's location. At present the technology has not developed to a state that it is possible to accurately measure a mobile devices movement/location. The popular GPS systems cannot yet identify the exact location of a device within a building as they typically measure a minimum of 3.5 m (Force, n.d.), thus making GPS sensors not attractive for the proposed application. Therefore we have manually measured the positions of the mobile device measurement points with respect to the light sources. Following the above process, illumination values at points x1, x2, x3 and x4 measured and their position information was recorded as (x_i, y_i).

In the experiments conducted the four light sources were placed at the edges of a square of side approximately 100 cm and all lighting sources were placed on a horizontal plane approximately 70 cm from the ground level. Switching the individual light sources on/off varied the illumination in the region where the experiments were conducted. Table 6.6 tabulates the illumination values captured at the four points under different lighting configurations.

Table 6.3 Illumination values for the four positions when using different lighting configurations

Scenario	X1	X2	X3	X4
A (all lights are on)	39	298	131	273
B (4 is off the rest are on)	28	273	129	267
C (3 is off the rest are on)	24	280	8	288
D (2 is off the rest are on)	34	7	125	294
E (1 is off the rest are on)	17	93	116	4
Bb (4 is on the rest are off)	3	0	1	0
Cc (3 is on the rest are off)	6	0	108	3
Dd (2 is on the rest are off)	0	89	1	0
Ee (1 is on the rest are off)	9	4	4	300
F (3 and 4 are off the rest are on)	11	104	5	306
G (1 and 2 are off the rest are on)	15	1	113	3
H (2 and 3 are off the rest are on)	19	6	9	313
I (1 and 4 are off the rest are on)	7	84	98	3
J (2 and 4 are off the rest are on)	23	4	116	338
K (1 and 3 are off the rest are on)	5	93	2	0

A closer investigation of the data provided in table 6.6 reveals the impact of illumination at a given point caused by each light source.

6.3.2 Experiment 2 – Environment Lit by Non-Symmetric Light Sources

This experiment was conducted using four light sources (small circles in figure 6.28) placed in a non-symmetric configuration. The lights were placed on tables, approximately 75cm from the ground level. The mobile device was placed on the floor facing upwards in seven different positions (points labelled as p1 to p7 in figure 6.28).

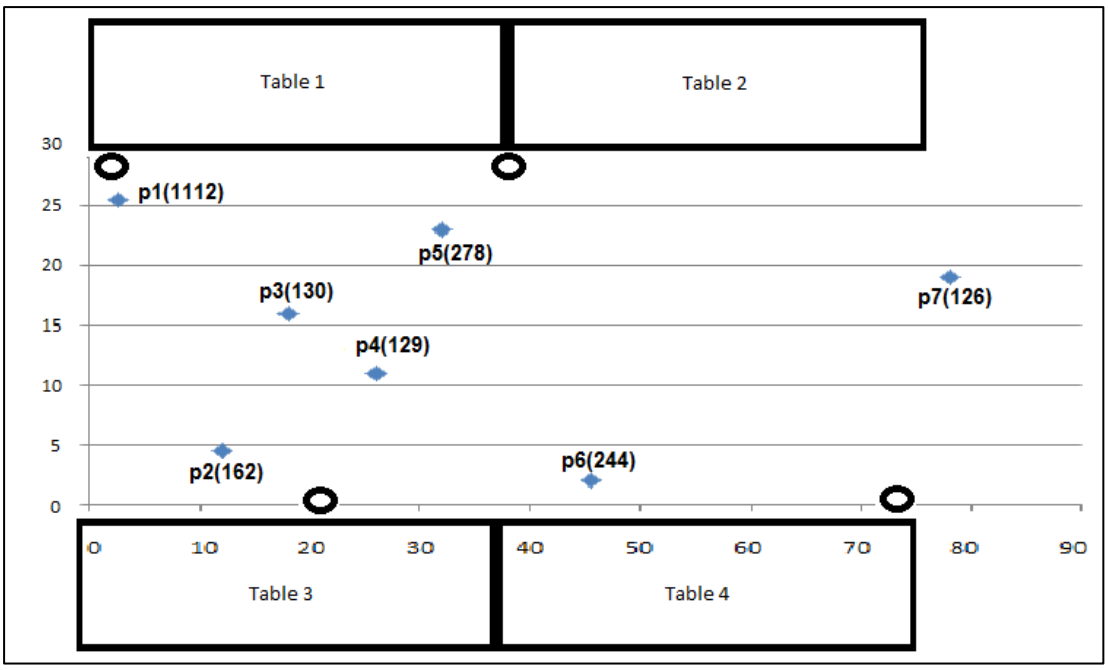


Figure 6.16 Experiment 2 - Non-symmetric light source configuration

After placing the mobile device at a particular point in order to capture the ALS readings as described above, the sensor data capture (ALS reading only in this case) software application was switched on to allow capturing illumination level at the given point caused by different light sources. The mobile device was kept for a couple of seconds in each position to capture a number of different readings, which were finally averaged to obtain the true illumination value at the point.

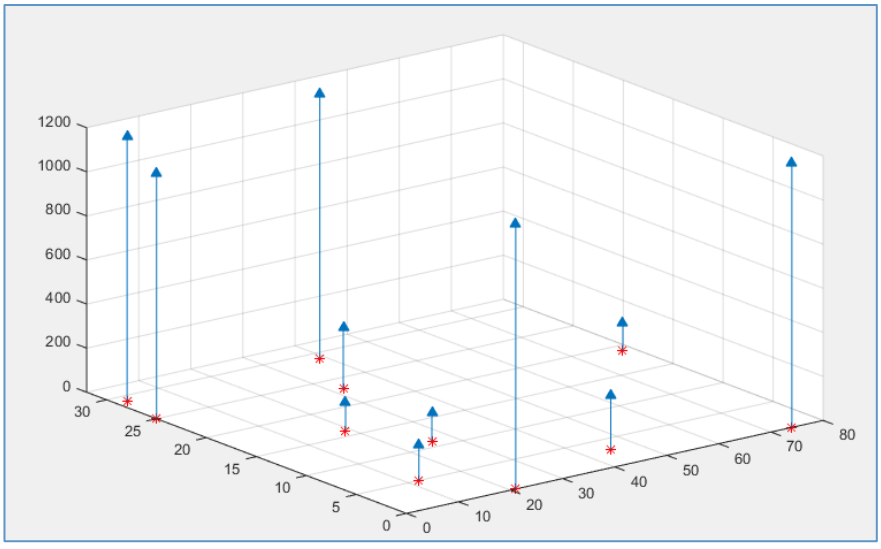


Figure 6.173-Dimentional representation of the seven positions of the phone in x-y plane and the illumination value in each position plotted along the z-axis

Due to the incapability of the mobile device to accurately and at high precision calculate the distance it moves within a small area of less than three and half meters, the exact positions of the phone was manually measured along an x, y and z-axis. However, it is assumed that in the near future, that there will be a possibility for identifying the exact position of the phone, within several centimetres of the true value. For all the experiments conducted within the context of research presented in this thesis, the exact positions were manually measured and made available.

After recording the illuminations variations that result from the two experiments above, it is vital to determine the possibility of a modelling the captured data with the hope of predicting the illumination value at any given point within the area being investigated. In the following sections we evaluate the use of Polynomial functions of various orders and the use of machine learning algorithms to accurately predict illumination.

6.3.2.1 Polynomial Model

Polynomial models have the flexibility of accurately fitting into complex data relationships, in particular when using the higher order terms. Therefore they are expected to perform more accurately than the Sum-of-Sine and Fourier Transform based techniques investigated in section 6.2.

Table 6.4 Evaluation of Performance of Polynomial Models

Degree		R-Square	Equation
x	y		
2	1	0.841	p00 + p10*x + p01*y + p20*x^2 + p11*x*y
3	1	0.8962	p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p30*x^3 + p21*x^2*y

4	1	0.9308	p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p30*x^3 + p21*x^2*y + p40*x^4 + p31*x^3*y
1	2	0.2996	p00 + p10*x + p01*y + p11*x*y + p02*y^2
1	3	0.4035	p00 + p10*x + p01*y + p11*x*y + p02*y^2 + p12*x*y^2 + p03*y^3
1	4	0.954	p00 + p10*x + p01*y + p11*x*y + p02*y^2 + p12*x*y^2 + p03*y^3 + p13*x*y^3 + p04*y^4
2	2	0.8638	p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2
3	2	0.9338	p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2 + p30*x^3 + p21*x^2*y + p12*x*y^2
2	3	0.9554	p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2 + p21*x^2*y + p12*x*y^2 + p03*y^3
3	3	0.9735	p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2 + p30*x^3 + p21*x^2*y + p12*x*y^2 + p03*y^3

The results tabulated in Table 6.8 indicate that a 3rd order polynomial of x and y demonstrated a prediction accuracy of 0.9735 which is the highest. Although higher degree polynomials would have improved the prediction beyond this they are computationally extensive to implement. Figure 6.30 graphically illustrates the third order polynomial model.

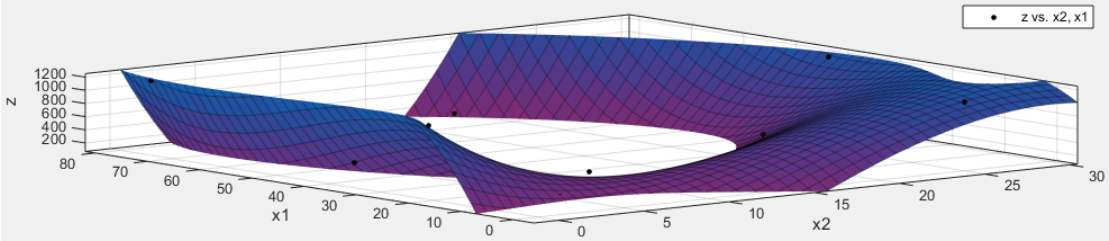


Figure 6.18 3rd Degree Polynomial Model

In chapter 5 it was proven that sum of sine is the best model to be used for one and two light sources environment. Consequently, it was used in chapter 6 as it was stated in section 6.2.1. Polynomial on the other hand, generally did not perform as well as sum of sine under one and two light sources. For this reason it was eliminated. However, higher polynomial orders have proven to give good accuracy rate in 2D illumination modelling. Seeking good accuracy rate and less complex models did not make polynomial as an option due to its complexity.

6.3.2.2 Machine Learning

Machine Learning algorithms (see Chapter 3.4.3.5) have been traditionally used to model complex distributions of data and are likely to provide means for more accurate models. In this section research was conducted to capture data as 10 and 100 values respectively from each point. The use of four Machine Learning algorithms was investigated for the prediction of illumination. The results are tabulated in Table 6.7. When more data values are measured at a single point of illumination capture and all data captured is used in the modelling, the predicted accuracy significantly increased. It is noted that Bagging with REPtree as the base learning algorithm has gained the best by having additional input data. With 100 values for each location, the accuracy obtained is significantly high in all tested models. It can also be concluded that even with 10 readings per point the accuracy figures have been sufficiently high to warrant their use in practice. Details explanation of the algorithms is mentioned in section 3.6.5.

Table 6.5 Different classifiers implemented for the seven positions in the non-symmetric experiment

Classifier	Correlation coefficient 10 values for each position of the phone (1 sec)	Correlation coefficient 100 values for each position of the phone (10 sec)
AdditiveRegression. DecisionStump	0.9382	0.9857
AdditiveRegression. M5Rules	0.9455	0.9998
Bagging. REPtree	0.7561	0.9999
M5Rules	0.9121	0.9997

6.4 Conclusion

The research conducted in this thesis has revealed that for more complex lighting patterns that are caused by increased number of light sources (regardless of the fact whether they are placed in regular or irregular patterns) and complex movement paths of mobile devices in the environment that deviate from a straight line, causes the simple mathematical models that were proven to be very effective in modelling illumination caused by simple lighting set ups (see Chapter 5), to fail. Although computationally efficient Sum-of-Sine functions performed well in modelling illumination of simpler environments, they failed in modelling illumination in more complex environments. Mathematically more costly functions such as higher order Fourier Series expansions were more effective than the Sum-of-Sine functions but performed sub optimally when the complexity of the environment increased. Higher order polynomial functions performed well in more complex environments but in order to maintain a reasonable level of prediction accuracy, the order of the polynomial had to be high, creating complexity in computational implementation.

Concluding the research finally Chapter 6 investigated the basic use of machine learning algorithms for the modelling of illumination in the most complex, practical lighting configurations. The popular machine learning algorithms tested provided a significant promise of prediction capability. Extending this research, in Chapter-7 a comprehensive study of optimal use of machine learning algorithms in modelling illumination is conducted.

Chapter 7 : Multi Dimensional Illumination Modelling & Prediction using ALS, Gyroscope and Accelerometer

7.1 Introduction

The research conducted in Chapter 6 concluded that predicting illumination in an environment lit up by a complex configuration of light sources and in particular when the user's motion pattern is complex, is a task that can only be potentially handled by machine learning algorithms. A number of mathematical functions tested, both in Chapter 5 and Chapter, 6 indicated that they are only suitable for modelling illumination in environments lit by a simple configuration of lighting source.

In the experiments conducted in Chapter 6 only the ALS readings measured at the data collection points spread across the environment were used in creating the model for illumination. At each point the mobile device was kept on a horizontal plane and readings were taken after leaving sufficient time for the ALS sensor value to be stable. Multiple ALS readings were taken at each point (see section 6.3.2.2) while holding the mobile device in a horizontal plane. These readings were fed into the above modelling process.

In a practical implementation of this illumination modelling approach within a mobile device one has to implement a mobile App that captures data in an initial calibration routine that will require the bearer of the mobile device to record data (knowingly or unknowingly) in multiple locations and multiple times at each location that can then be used in a secondary stage that creates the model. The need to wait for the ALS reading to be stabilized is due to the fact that one has to compensate for the latency of operation of the ALS. Once the model is created it is then possible to use it to predict the stable ALS reading value that one would expect to obtain if the mobile is moved to any required (x, y) position in the test plane. This predicted value will be a quick estimate of the true ALS reading at that point, that the ALS reading will finally arrive at, once the mobile

device is moved to the point and sufficient time is given for the sensor reading to be stabilized. The practical challenge behind the above data collection, modelling and prediction process is that most practical usage scenarios of a mobile device will not satisfy the assumptions based on which the above process is built on. In particular a user cannot be expected to stay still holding the phone at few points to allow the ALS readings to stabilize. This is the case in particular if the calibration process is carried out automatically without the user knowing or the user being strictly instructed about the calibration routine.

As proven in Chapter-4, the accelerometer and gyroscope values at a measurement point give a valuable insight into the conditions in which the data collection is done for modelling and also the general behaviour of the user at the time of the model needing to produce a predicted illumination value. For example the gyroscope and accelerometer readings will indicate the deviation of the phone from a horizontal plane, i.e. tilting the phone, holding the phone and sitting down from a walking position etc. It is noted that if the phone is tilted by any given amount as compared to holding it flat (i.e. horizontally) the ALS readings will change as the lighting configuration seen by the mobile device's ALS will change even though the physical lighting configuration has not changed. If the phone is moved up or down in a vertical direction, once again such changes in illumination, will occur. Therefore incorporating a mobile device's behaviour analysis to aid in the process of illumination prediction can be recommended.

Given the above observations in this chapter in addition to the ALS reading, we also make use of the accelerometer and gyroscope readings (along x, y and z axes) to model and predict the illumination value. The intention is to make the data capture, ALS value modelling and prediction processes more practically relevant by considering the mobile device's motion, rotation etc. For instance, if the user is moving in a straight line under a light source without rotating the phone, this will result in having a steady increment on the illumination values as it has been proven in chapter 5. However, in more real life scenarios, the user does not hold the phone still, instead he/she rotates the phone in different directions while standing or walking. Fusing or combining the data of the

accelerometer and gyroscope will result in the ability to identify the rotation angle of the phone with respect to the position of the light source. This will consequently help in predicting the illumination values based on the rotation of the phone in the environment it is present in.

For clarity of presentation this chapter is divided into several sections. Apart from this section, which introduced the limitations of the modelling and prediction process proposed in Chapter 6 and the need for a practical solution, section 7.2 formally defines the research problem. Section 7.3 provides details of the experiments conducted and section 7.4 analyses the results obtained in detail. Finally section 7.5 concludes with an insight to future work.

7.2 Research Problem

Assume that the user is sitting in a given position within a room where the lighting configuration remains fixed. In such an environment the illumination reading recorded by the ALS at any given point will be fixed, provided that the user holds the mobile device stationary/still. Any change of the position of the mobile device or the angle at which it is held will result in a change of ALS reading. The amount of change will depend on the movement, translational or rotational.

In a typical practical situation, in which the mobile device user is attempting to observe the device's screen, some level on movement of the mobile device should be tolerated. Otherwise the practical relevance of the illumination model can be challenged. For examples while using the mobile device, whilst in one single position in the room, the user may still change the position of the mobile device for instance by sitting/standing or bringing the phone closer or further from his/her eyes. The user may also change the orientation/angle of the mobile whilst stationary, in an attempt to observe the content in the screen more clearly. In either case the illumination as seen by the ALS will change, and this needs to be factored into the model design to obtain a true 3D illumination model. Further obvious changes to the illumination will happen if the mobile

device user moves to a different position in the room. A ALS reading prediction model that can cater for all the above changes will greatly enhance the user experience when integrated with mobile Apps that are highly sensitive to illumination variations, as described in Chapter-1, overcoming the key limitations of a typical ALS, i.e. latency and narrow angle of operation.

The mobile devices position and behaviour dependent illumination can be modelled by measuring its relative location in 3D space (i.e. x, y and z), its accelerometer (i.e. in x, y and z directions) and gyroscope (i.e. in x, y and z directions) readings. Thus we make use of the mobile App of section 3.5 to gather the above data.

The illumination modelling that results from the above will be multi-dimensional, i.e. in a an environment of a fixed configuration of lighting sources the illumination can be modelled at points in 3D space, but at each point the model can also withstand rotations around any of the phone's axes, i.e. x, y and z.

7.3 Experiments

Experiments were conducted in two phases. In the first phase the phone was kept in a single horizontal plane at all times, and rotated around the x-axis only. In the second phase, the first phase's experiments were repeated in two additional horizontal planes, but were also rotated around x and y planes. In all cases the (x, y, z) readings of the mobile device's location, accelerometer and gyroscope readings were recorded alongside the ALS readings. Note that no rotations were done around the Z axis of the phone as due to the 'cone' shaped nature of the ALS response pattern (see figure 3.4) a rotation around the z axis is not going to cause any changes to the ALS reading.

For all of the experiments the light sources were approximately set up in a single horizontal plane and were equally and regularly spaced. The configuration mimics the lighting configuration of a typical indoor environment. It is however noted that the proposed modelling and prediction approach can be also applied

to model any general lighting configuration and it is not essential that the light sources are placed in a single horizontal plane.

7.3.1 Phase 1- Illumination Modelling in a Single Horizontal Plane

In this experiment the phone was placed in 14 different points of a single horizontal plane, as illustrated in figure 7.1. The test data points were selected in a way so that some are close to the light sources while others are not as close. It was also ensured that the test data points were randomly spaced within the said environment (see figure 7.1, the 'diamond' shaped depicted points).

In each of the positions, whilst the mobile device was held fixed, the mobile App captured the raw data values from the ALS, Gyroscope and Accelerometer. Further the (x, y, z) location of the test point with respect to the lighting configuration was also measured and recorded. Four rotation/tilt angles were considered for each point, 0° (flat), 10°, 20°and 30° (rotation around x axis). Although having more points would increase the accuracy of the resulting model, our focus is also to maintain the simplicity of the model creation process and hence to only consider readings at the above four angles. For each point and each angle, 30 readings are taken to compensate for signal noise in the accelerometer and gyroscope.

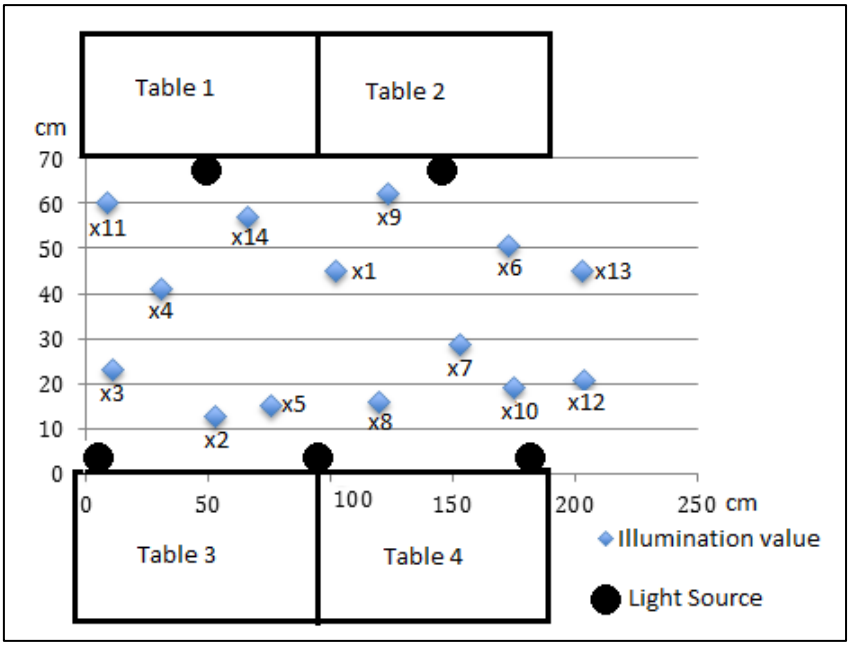


Figure 7.1 Phone positions in Phase-1 experiment

Table 7.1 tabulates a sample of data values for location X1, indicating the ALS, Accelerometer, Gyroscope values and measured angular tilt around the x-axis. Note that the rotation angle labels for point X1, depicted as, x1ZeroAngle, x1TenAngle, x1TwentyAngle and x1ThirtyAngle represents rotations 0°, 10°, 20°and 30° respectively around the mobile device's x-axis. Note that this column of labels does not enter the modelling process as the rotational movement will be represented by the accelerometer reading in the y direction (see Figure 7.2). The plots of figure 7.2 reveals that as the phone is held stationary at the time that the measurements are taken at each different angle, the only discriminant value is the accelerometer-y value as the different angle creates a component (dependent on the angle) of the gravitational acceleration present, to impact this measure.

Table 7.1 Sample of the raw data for the ALS, Accelerometer and Gyroscope

Illumination (lux)	Accelerometer X Axis	Accelerometer Y Axis	Accelerometer Z Axis	Gyroscope X Axis	Gyroscope Y Axis	Gyroscope Z Axis	Rotation around x-axis
::::::	::::::	::::::	::::::	::::::	::::::	::::::	::::::
315	0.2634	0.0646	9.3440	-0.0024	0.0061	0.0611	x1ZeroAngle
315	0.2634	0.0646	9.3440	-0.0024	0.0061	0.0611	x1ZeroAngle
316	0.2723	0.0575	9.3721	0.0018	0.0052	0.0580	x1ZeroAngle
316	0.2723	0.0575	9.3721	0.0018	0.0052	0.0580	x1ZeroAngle
::::::	::::::	::::::	::::::	::::::	::::::	::::::	::::::
306	0.2227	0.8110	9.2817	0.0058	0.0034	0.0589	x1TenAngle
307	0.2221	0.8074	9.2769	0.0021	0.0024	0.0553	x1TenAngle
307	0.2221	0.8074	9.2769	0.0021	0.0024	0.0553	x1TenAngle
314	0.2358	0.8200	9.2973	0.0040	0.0058	0.0565	x1TenAngle
::::::	::::::	::::::	::::::	::::::	::::::	::::::	::::::
299	0.2263	1.7765	9.1895	0.0034	0.0061	0.0577	x1TwentyAngle
299	0.2263	1.7765	9.1895	0.0034	0.0061	0.0577	x1TwentyAngle
315	0.2340	1.7771	9.1925	-0.0012	0.0058	0.0596	x1TwentyAngle
333	0.2292	1.7849	9.1680	0.0034	0.0043	0.0586	x1TwentyAngle
::::::	::::::	::::::	::::::	::::::	::::::	::::::	::::::
309	0.1921	2.4846	9.0220	0.0031	0.0034	0.0583	x1ThirtyAngle
318	0.1927	2.4882	9.0315	0.0027	0.0055	0.0550	x1ThirtyAngle
315	0.1927	2.4810	9.0537	0.0027	0.0055	0.0583	x1ThirtyAngle
315	0.1927	2.4810	9.0537	0.0027	0.0055	0.0583	x1ThirtyAngle
::::::	::::::	::::::	::::::	::::::	::::::	::::::	::::::

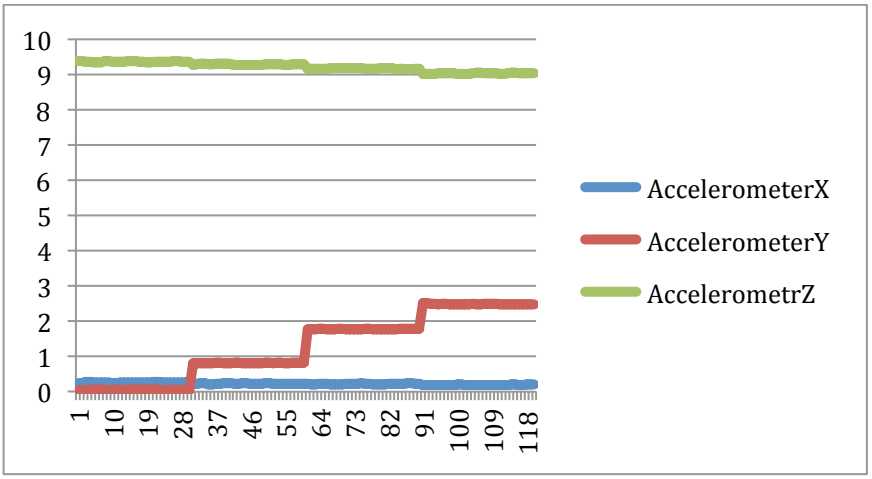


Figure 7.2 The Accelerometer readings when the phone is held at different angles (rotated around the X axis only)

7.3.2 Phase 2- Illumination Modelling in a 3D Space

In this experiment in addition to the data collected in section 7.3.2, two further sets of data of similar nature, but for only seven locations each, only, were captured at two different heights from the plane in which the data for section 7.3.2 were captured. Thus in this experiment the position of the mobile phone is modelled in 3D space of the environment in which the lighting sources are present. Each position is now defined as a point, (x, y, z). It is noted that the lighting sources and configurations for Phase-1 and Phase-2 experiments were identical. At each point the stable ALS readings were captured at three angles, namely when the phone is kept flat, in a horizontal position and when the phone is rotated around the x and y by +30 and –30 degrees. Note that for compensating for signal noise, 30 readings for each angular placement were captured and subsequently used in experiments.

Figure 7.3 illustrates the placement of all test points in the 3D space of the environment. Note that to illustrate the test points in the three different planes, different symbols have been used. The points in all three planes have been spaced randomly.

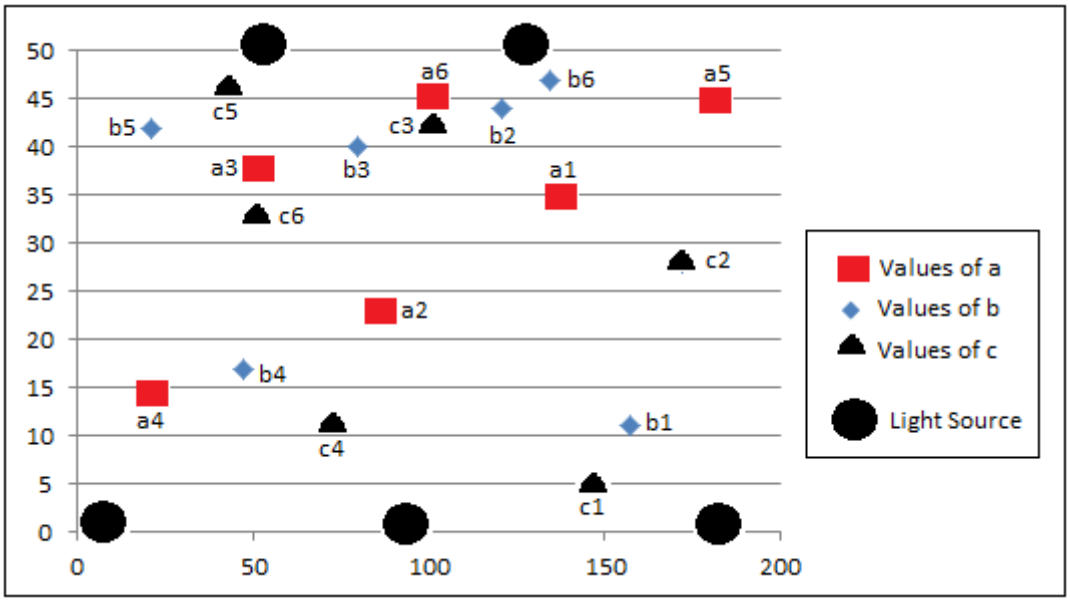


Figure 7.3 Phone positions in Phase-2 experiment: positions marked as, 'a', 'b' and 'c', represent respectively points at ground level, 6 cm and 12 cm above ground level

Figure 7.4 illustrates the potential changes that are introduced to the accelerometer readings when the measurements are taken at three different angles each (-30, 0 and +30) for rotation around x and y-axes. It clearly reveals that the Accelerometer X and Y values can be used to discriminate between the three different angles at which the data was recorded. The Accelerometer Z values do not play and discriminative role in this experiment.

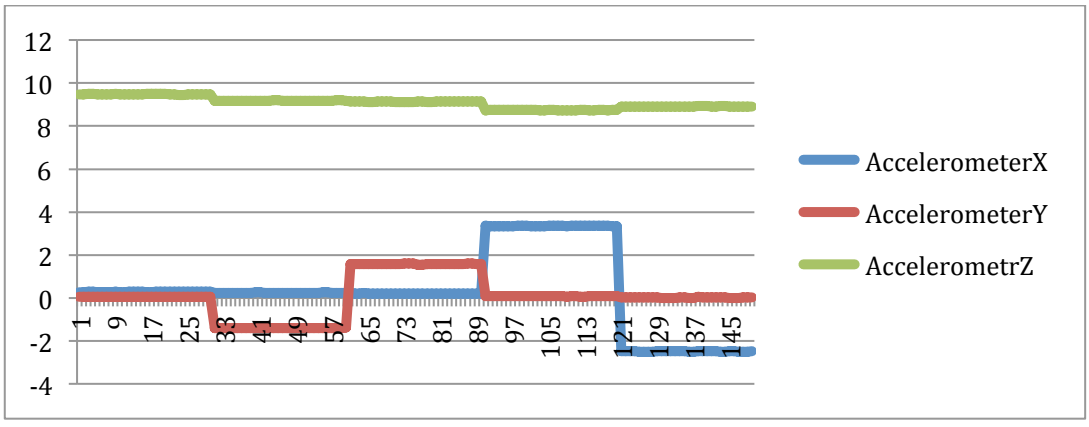


Figure 7.4 The Accelerometer readings when the phone is held at different angles (rotated around the X and Y axes only)

In figure 7.4, the time intervals approximately between 0 to 29, 29 to 60, 60 to 90, 90 to 120 and 120 to 150 represent the phone when it was kept flat, rotated to the left along the Y-axis, rotated to the right along the Y-axis, rotated along the X-axis, having the top side down and rotated along the X-axis, having the top side up respectively.

7.4 Results and Analysis

The captured raw data from the experiments in section 7.3.1 and 7.3.2 are transferred to a computer and saved in a .csv file and are subsequently used within Weka software tool for further data analysis. Sections 7.4.1 and 7.4.2 present a detailed analysis of the results obtained.

7.4.1 Phase-1

Having noted the (x, y) coordinates of the points of illumination capture, spread across a single horizontal plane, the gyroscope and accelerometer readings in (x, y, z) directions, in the following sections we experiment with using different models to predict illumination.

7.4.1.1 Scenario 1

In this scenario data collected from all of the 14 test locations were used in the modelling. Table 7.2 tabulates the prediction accuracies of Illumination models that have been generated using six Machine Learning algorithms. Considering the correlation coefficients obtained for the different models, the K-nearest Neighbour and Decision Table algorithms perform best. The Multilayer Perceptron algorithm performs the worst (giving a correlation coefficient of 0.8157) but it has been identified that this is due to the fact that the approach used the default setting for the number of hidden parameters. When the number of hidden layers was increased from 3 to 12, the correlation coefficient improved to 0.9837.

It is noted that in the above analysis, the three components of the gyroscope (i.e. the components in x, y and z directions) and the z component of the accelerometer remain redundant parameters in the determination of the model.

This is due to the fact that these values remained generally constant throughout the data collection process.

Table 7.2 Prediction accuracy results for Phase-1 experiments

Machine Learning Algorithms	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error
K-Nearest Neighbour	0.9999	2.2857	9.6479	0.5888%
REPtree	0.9821	23.5766	113.7219	6.0729%
Bagging. REPtree	0.9869	22.3711	97.7034	5.7623%
Random Subspace. REPtree	0.9934	27.4066	71.5678	7.0594%
DecisionTable	0.9988	9.4919	28.9284	2.4449%
MultilayerPerceptron	0.8157	249.6611	364.5692	64.3075%

The above results for the Phase-1 experiments reveal that the illumination modelling on a given horizontal plain within an environment lit up by multiple light sources can be carried out accurately using machine-learning algorithms. Relative absolute error indicates approximately 60% difference between MultilayerPerceptron and the rest of the algorithms. A possible reason to this high number of errors is that MultilayerPerceptron was used with its default parameters. However, changing the parameters would reduce the relative absolute error and would result in complicating the algorithm. The nature of the data along with the simplicity of the K-Nearest Neighbours have possibility reduced the error rate compared to the MultilayerPerceptron algorithm.

In the following sections we analyse the impact of using less measurement/training points on the model accuracy. We also study the impact of not spreading the measurement/training points evenly within the 2D plane. Instead of taking the measurements again, in the following sections we select a subset of data points captured under the experiment described in section 7.3.1 for the modelling,

7.4.1.2 Scenario 2

In this scenario, only data from three points, namely, a point located at the far end of the left side (x3), a point in the centre (x1) and a point slightly to the right side (x7) were used in creating the model. Note that data captured at these points with the mobile device placed at all four angles, were taken into consideration in modelling. Further for each test point and phone orientation, all 30 readings initially recorded were taken into consideration. The purpose of this experiment was to determine the impact of using less data in creating the models. The same machine learning algorithms were used in the modelling and the results are tabulated in table 7.3.

Table 7.3 Impact of using less data points for illumination modelling

Machine Learning Algorithms	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error
K-Nearest Neighbour	0.9994	2.1861	7.1653	1.1544%
REPtree	0.9977	9.2493	14.2279	4.8844%
Bagging. REPtree	0.9978	8.5675	13.9927	4.5243%
Random Subspace. REPtree	0.9974	10.5301	15.713	5.5607%
DecisionTable	0.9947	9.6798	21.4475	5.1117%
MultilayerPerceptron	0.9973	11.5957	15.5264	6.1234 %

The results tabulated in table 7.3 reveal negligible impact on the accuracy of the models when the number of data points is reduced as low as three. Once again K-NN remains the algorithm with the highest accuracy. The accuracies of all other algorithms have increased except for RandomSubspace.REPtree whose accuracy has slightly dropped.

7.4.1.3 Scenario 3

In this experiment only 20 of the 30 sets of readings obtained at a given point/orientation were considered in the modelling process. This amounts to a total of 80 illumination values per test location. As in scenario-2, only data measured at the three positions x1, x3 and x7, were considered.

Table 7.4 Impact of using fewer readings per measurement point

Machine Learning Algorithms	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error
K-Nearest Neighbour	0.9988	3.2667	10.307	1.7216%
REPtree	0.9977	9.4861	14.119	4.9993%
Bagging. REPtree	0.9982	8.3822	12.5736	4.4175%
Random Subspace. REPtree	0.9971	11.0682	16.2885	5.833%
DecisionTable	0.9961	9.7953	18.5594	5.1622%
MultilayerPerceptron	0.9964	14.2682	18.1441	7.5195 %

The results tabulated in table 7.4 reveal that reducing the number of different readings taken at a test point slightly decreases the accuracy of the models obtained. This is expected as the lack of data for training will result in less accurate prediction models.

7.4.1.4 Scenario 4

In this test scenario the data used in the test scenario-2 were used but with the exception of data collected at each point oriented at angles 10° and 20°.

Table 7.5 Impact of using data from a fewer number of angular orientations

Machine Learning Algorithms	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error
K-Nearest Neighbour	0.9994	1.9417	7.357	1.0342%
REPtree	0.9986	7.383	11.0231	3.9324%
Bagging. REPtree	0.9987	6.833	10.4033	3.6394%
Random Subspace. REPtree	0.9975	10.6517	15.0327	5.6734%
DecisionTable	0.9984	8.0284	11.8157	4.2761%
MultilayerPerceptron	0.9977	9.9035	13.9463	5.2749 %

The accuracies obtained remain high indicating that measurements need not be taken at smaller angular intervals, to collect data for accurate modelling.

Table 7.6 summarizes the four models developed above and the results thus obtained, when using different classifiers. The accuracies of all experiments are indicated only with respect to the Correlation Coefficient.

Table 7.6 Summary of results: modelling illumination in a 2D plane using all data gathered at a given test point

No. of Positions	Angles Of the Phone	No. Illumination values captured	Correlation coefficient					
			K-Nearest Neighbour	REPtree	Bagging. REPtree	Random Subspace. REPtree	Decision Table	Multilayer Perceptron
14	0, 10, 20, 30	30	0.9999	0.9821	0.9869	0.9934	0.9988	0.8157
3	0, 10, 20, 30	30	0.9994	0.9977	0.9978	0.9974	0.9947	0.9973
3	0, 10, 20, 30	20	0.9988	0.9977	0.9982	0.9971	0.9961	0.9964
3	0, 30	20	0.9994	0.9986	0.9987	0.9975	0.9984	0.9977

From the results summarized in table 7.6 it can be concluded that the accuracy of the prediction models heavily depend on the number of readings taken at a given measurement position/orientation. This is justified by the fact that as the predictions are done using raw-data values from the gyroscope and accelerometer, the larger number of measurements taken at a single position/orientation can lead to the removal of the impact of noise and sensor latencies on the models created, thus giving higher prediction accuracy when larger number of measurements are taken at any given test point.

It is noted that in all the above experiments the (x, y, z) values of both the gyroscope and the accelerometer were used despite the fact that apart from the x direction value of the accelerometer, all other values remained rather constant and hence redundant, in the modelling process. Table 7.7 tabulates the results of

modelling based on 14 test points but with all of the accelerometer and gyroscope values removed from the modelling process.

Table 7.7 The impact of not using the accelerometer and gyroscope values in the modelling process

Machine Learning Algorithms	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error
K-Nearest Neighbour	0.9655	72.5455	157.1035	18.6862 %
REPtree	0.9655	72.5711	157.1119	18.6928 %
Bagging. REPtree	0.9655	72.5416	157.0487	18.6852 %
Random Subspace. REPtree	0.9651	76.4279	157.9425	19.6863 %
DecisionTable	0.9655	72.5455	157.1035	18.6862 %
MultilayerPerceptron	0.1921	403.8837	599.6486	104.032 %

The results tabulated in table 7.7 indicate that the accuracies have dropped significantly as compared to that obtained under scenario-1, showing the importance of at least using the accelerometer x value in the modelling process. To model the impact of the rotational behaviour of the mobile device the accelerometer readings are essential. This aspect is further investigated in the analysis carried out in the next section where illumination prediction models are developed to be used in a 3D plane.

In table 7.7 it is specifically noted that the Multilayer Perceptron performs poorly. This is due to the fact that the default number of hidden layers (three in total) has been used to obtain the relevant modal. If the number of hidden layers is increased to 12 the correlation coefficient increases to 0.6746.

7.4.2 Phase-2

The experiment carried out to gather data for the analysis presented in this section was described in section 7.3.2. The experiment aims at modelling

illumination in a 3D environment covered by the data collection points that were spread on three separate 2D planes. Approximately 6 measurement points were considered in each of the planes. At any given measurement point, the mobile phone was rotated in both the x and y planes separately to collect accelerometer and gyroscope readings at two different angles, i.e. at 0^0 and 30^0.

Figure 7.8 tabulates prediction accuracy obtainable by using different machine learning algorithms when the accelerometer and gyroscope values were not used in the modelling. The prediction accuracy obtainable is significantly lower that the prediction accuracy obtained when accelerometer and gyroscope values were ignored in the modelling of 2D illumination. It is noted that in the 3D illumination modelling experiment the mobile phone was rotated both around the x and y axis, hence making the illumination changes resulting from such a movement more complex and difficult to manage. It is also noted that a close look at the accelerometer and gyroscope raw data readings obtained revealed that the three gyroscope values remain largely constant. As the phone was not moved laterally at each measurement point the gyroscope readings will remain insignificantly small. This is due to the fact that there is no change of velocity of the device in any of the directions. As the phone was not rotated along the z-axis the accelerometer z direction readings recorded were also insignificant.

Table 7.8 Prediction accuracy results for Phase-2 experiments with accelerometer and gyroscope readings ignored in the modelling process

Machine Learning Algorithms	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error
K-Nearest Neighbour	0.9311	71.2396	135.6814	24.8819 %
REPtree	0.931	71.3262	135.7881	24.9122 %
Bagging. REPtree	0.9313	71.0708	135.4704	24.823 %
Random Subspace. REPtree	0.9312	71.2816	135.5556	24.8966 %
DecisionTable	0.9311	71.2396	135.6814	24.8819 %
MultilayerPerceptron	0.6249	206.9033	293.026	72.2653 %

Table 7.9 tabulates the prediction accuracies obtained when all of the accelerometer and gyroscope values were considered in the modelling of illumination. The prediction accuracies have improved significantly and have come in line with the 2D illumination prediction accuracies obtained in section 7.4.1 when all data points, 30 readings per point, and all values of the accelerometer and gyroscope values were considered. It is also noted that the Multilayer Perceptron also performs significantly well even with the default number of hidden layers, given the fact that a significant number of data points are used in the modelling process.

Table 7.9 Prediction accuracy results for Phase-2 experiments with accelerometer and gyroscope readings considered in the modelling process

Machine Learning Algorithms	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error
K-Nearest Neighbour	0.9998	1.4401	6.771	0.503 %
REPtree	0.9982	9.6658	22.3973	3.376 %
Bagging. REPtree	0.9988	8.8181	18.2124	3.0799 %
Random Subspace. REPtree	0.9965	17.2248	33.3692	6.0161 %
DecisionTable	0.9992	5.9606	14.871	2.0819 %
MultilayerPerceptron	0.97	67.8971	90.5215	23.7145 %

The results tabulated in table 7.9 proves that all of the machine learning algorithms considered for modelling the illumination within a 3D space of a mobile device is capable of predicting illumination values effectively. A calibration phase during which few measurements are taken at random points of the 3D space is needed for training such a model. Once a model has been trained it will be able to predict illumination accurately.

In the above experiment we considered data recorded at approximately 18 points for training the model. If half of these points were used in the modelling process the accuracy obtainable from the K-Nearest Neighbour algorithm did not reduce significantly and remained at 0.8434.

7.5 Conclusion

Experiments conducted in this chapter have proven that it is possible to use machine-learning algorithms to accurately predict the illumination variation in a 3D space of use of a mobile device in a lit up area.

We considered a regular configuration of light sources spread on a horizontal plane but argued that it is not a requirement that the methods adopted could only predict illumination in an environment of regularly placed light sources. Rather our experiments attempted to mimic a lighting configuration often present in within an indoor environment.

The K- Nearest Neighbour algorithm produced the best results followed closely by four other machine learning algorithms tested including the popular Multilayer Perceptron (Neural Network). In the case of optimally using the Multilayer Perceptron based approach it was found out that it is vital to select the right number of hidden layers for the algorithm to perform optimally.

The analysis carried out in different ways revealed the importance of using the accelerometer and gyroscope values. As in the case of the experiments conducted the illumination measurements were taken at fixed points with the phone stationary (although the phone could be rotated around the x, y and z axes) the gyroscope readings were insignificantly small and could be excluded from the modelling. However as illumination values were captured whilst the phone was being rotated, placed at these fixed points (this mimics the normal use of a mobile phone by a user who is sitting or standing in one place), the appropriate accelerometer readings had to be included in the modelling. Note that the angular placement of the phone axis results and the gravitational acceleration playing a role in the accelerometer readings which has been

successfully exploited in the proposed models to model the variation of illumination that results from rotating the phone whilst being in a fixed position in an environment lit up by a fixed set of illumination sources.

Given the above, this chapter has successfully demonstrated the possibility of modelling the illumination within a 3D space of a mobile device that is being normally used, aided by the accelerometer readings. The model is generated through a calibration stage where sample ALS readings and accelerometer readings are captured and used at training datasets to train a machine learning algorithm that will subsequently be able to predict illumination in any given location within the said space. This application can be used to resolve the important challenges the present day mobile devices face due to the limited angle of operation and latencies of the ambient light sensors. The chapter combines the knowledge gathered in chapters 4, 5, and 6 to provide the solution to the said problem.

In chapter 7, fusing the data of three sensors helped in resolving the two drawbacks of the ALS. Machine Learning algorithms were used to predict the illumination values of the ALS before the ALS is capturing them.

Chapter 8 : Conclusion

8.1 Conclusion

The research conducted within this thesis concluded that advance Machine Learning algorithms can effectively be used to seamlessly model the illumination variations within a three dimensional space that is lit up generally by any configuration of fixed light sources. Even though a mobile handheld device has not been originally designed to carry out illumination modelling around its space, due to the presence of a number of in-built sensors that measure the device's motion characteristics and a sensor that is capable of measuring ambient illumination in the surrounding space, it will be possible to make use of the outcomes of research presented in this thesis to enable this capability. As the illumination variation as seen by a device, not only depend on the lighting configuration but also on the device's motion characteristics, an ultimate model should take into account both the static illumination of a point in the 3D space and also the dynamic nature of the device's movement. A unique contribution of the work presented in this thesis is the combination of the above two aspects in accurately modelling illumination as seen by the device, taking into account the users intent and/or phone's usage, thus making the system suitable for systems that can enhance user experience.

The above capability provides a solution to two existing shortcomings of the Ambient Light Sensor of a mobile device, i.e. their inability to act sufficiently quickly (i.e. operational lag) and their narrow angle of operation. It was mentioned that these two shortcomings have an adverse impact on the global, multi-award winning commercial patent, Iridix™, being challenged on its potential ability to resolve an open research problem that the state-of-the-art technology is at present unable to solve, i.e. effectively, dynamically and real-time adapt a screen's backlight illumination depending on the illumination of the surroundings of a mobile device, in order to provide a potential mobile device viewer with the best possible user experience.

Although the research conducted within the scope of this thesis was driven by a commercial need, it addressed an open research problem. Previous research has conducted investigations into multi-sensor data fusion that adopted statistical, mathematical and machine learning based approaches. However such work was limited to direct recommendations being used for the use of a named mathematical model or a machine learning based approach. Such studies were not rigorous in nature and the implementation complexities often prohibited the practical use of some of the algorithms proposed. The complexity of algorithms was not considered in the investigations when any recommendations were made.

Our detail review of literature indicated that there was no research conducted into modelling illumination within a 3D space, not alone the use of machine learning algorithms to solve the problem concerned. Therefore the research presented in this thesis that proves the concept that illuminations variations within a 3D space can be modelled and such models can be used in prediction of illumination in previously unknown locations, is a significant contribution to research. Combined, the research supported providing a practical solution to a unique but an open, applied research problem that promises significant impact to not only mobile device manufacturers but also their enthusiastic users.

In chapter-4 the use of a number of different machine learning algorithms to fuse together the raw data captured by both the Accelerometer and Gyroscope sensors, was investigated. Both single and ensemble learning algorithms were considered. The research concluded that in general the K-NN algorithm performed the best, overall. However, ensemble-learning algorithms such as Random Forests and Bagging with Random Forest as the base learner also performed very well. The research concluded that the differentiating the user 'sitting' and 'standing' but still in both cases, was very different as both sensors were generating very similar readings along all three axes. It was also revealed that removal of noise from the raw data could help in marginally improving the results in the case of using some particular learning algorithms.

The research conducted in Chapters 5,6 and 7 built a solution step-by-step to the problem of illumination modelling. Different lighting configurations, the use of both mathematical models and machine learning algorithms, different movement patterns of the mobile device, etc., were studied and their impact on perceived illumination by a device was investigated in detail. Simpler mathematical models such as Sum-of-Sine performs reasonably accurately modelling illumination in environments lit up by simple lighting configurations but when the complexity of configurations increased, more computationally costly mathematical models such as Fourier Series and Higher Order Polynomial functions had to be used. For the most complicated lighting configurations and device motion patterns, only machine learning algorithms performed well. All mathematical models failed to give reasonably accurate predictions.

The research presented on this thesis can be implemented an integrated within a mobile device so as to enable the mobile handheld device to act to illumination changes and user intent, more rapidly. At present a commercial implementation is being carried out to support this activity.

In the research conducted in this thesis the machine learning was done on raw data. The reason for this was to keep the computational cost of the system to a minimum. It will be interesting to determine if feature extraction from raw data and using such features for learning can improve accuracy of the models.

Although the entire research project was carried out using first a Nexus-7 tablet and then a Samsung Galaxy S4 mobile device, the same framework can be applied to any other phone. However the models obtained will be different due to differing characteristics of the sensors. A re-design of the model for each make-model of the mobile phone will be needed.

Fusing the data of multiple sensors namely accelerometer, gyroscope and ALS along with the use of Machine Learning algorithms enabled high accuracy prediction of illumination. Machine Learning algorithms have proven to be very good in predicting illumination values by giving high accuracy rates results.

Implementation of machine Learning algorithms is less complex compared to mathematical models which makes it simpler in implementing and takes less processing time. This allows it to instantly provide prediction values, which will be implemented on the screen before the actual illumination values is being captured by the ALS. As soon as the ALS captures the value, the brightness will be adjusted accordingly.

8.2 Future Work

Chapter 7 presents the final work of the thesis where it combines three sensors namely the ambient light sensor, accelerometer and gyroscope to enable predicting the illumination of the environment where the handheld device is present. The improvement of the technology can result in extending this work by adding more advanced sensors to provide more accurate predictions. On the other hand enhanced less complicated Machine Learning algorithms can be implemented, tested and computational time considered seeking less complex algorithms and high accuracy rates.

Reference

Vishay.com (2012). Optical Sensors. [Online] Available at: http://www.vishay.com/docs/84154/appnotesensors.pdf [Accessed 24 Nov, 2014].

Anjum, A. and Ilyas, M. (2013). Activity Recognition using Smartphone Sensors. In: *10th Consumer Communications and Networking Conference.* Las Vegas: IEEE, pp 4-5.

Criminisi, A., Salnikov, E. and Sharp, T. (2012). Multi-Modal tone-mapping of Images. 8,290,295.

Awan, M., Guangbin, Z. and Kim, S. (2012). Activity Recognition in WSN: A Data-driven Approach. In: 7th International Conference on *Computing and Convergence Technology*. Seoul: IEEE, pp 15–20.

Aydemir, O. and Kayikcioglu, T. (2013). Comparing common machine learning classifiers in low-dimensional feature vectors for brain computer interface applications. *International Journal of Innovative Computing, Information and Control*. *9*(3). pp 1145–1157.

Cheng, C. and Bai, Y. (2012). Using fuzzy logic and light-sensor for automatic adjustment of backlight brightness in a mobile computer. In: 16th International Symposium on Consumer Electronics. PA: IEEE, pp 2-3.

Boor, C. (1978). *A Practical Guid to Spline*. New York. Springer-Verlag

Breiman, L. (1996). Bagging Predictors. [Online] Carlifornia: University of California. p 1-4. Available at : http://www.cs.utsa.edu/~bylander/cs6243/breiman96bagging.pdf [Accessed 14 Jul. 2016].

Coskun, D., Incel, O. and Ozgovde, A. (2015). Phone Position / Placement Detection using Accelerometer : Impact on Activity Recognition, (April), 7–9.

Dietterich T.G. (2000) Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, vol 1857. Springer, Berlin, Heidelberg

Display Mate. (2010). BrightnessGate for the iPhone & Android Smartphones and HDTVs. [Online] Available at: February 23, 2015, from http://www.displaymate.com/AutoBrightness_Controls_2.htm [Accessed on 23 Feb, 2015]

MathWorks. (1994). Evaluating Goodness of Fit - MATLAB &amp

http://uk.mathworks.com/help/curvefit/evaluating-goodness-of-fit.html [Accessed on 26, Aug 2015]

Force, U.S.A. (2015). GPS Accuracy. http://www.gps.gov/systems/gps/performance/accuracy/ [Accessed on 29, Sept 2016]

Greg Milette, A. S. (2012). *Professional Android Sensor Programming*. Indiana: p66-89

Guo, H. (2011). A simple algorithm for fitting a gaussian function. *IEEE Signal Processing Magazine*, *28*(5), 134–137. doi:10.1109

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844

Holmes, G., Hall, M. & Frank, E. (1999). Generating rule sets from model trees. (Working paper 99/02). Hamilton, New Zealand: University of Waikato, Department of Computer Science.

Ian H. Witten, Eibe Frank, M. A. H. (2011). *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers.

Jayanthi, S. K., & Sasikala, S. (2013). Reptree Classifier for Identifying Link Spam in Web Search Engines. *Ictact Journal on Soft Computing*, *3*(02), 498–505. http://ictactjournals.in/paper/IJSC(Jan2013)_Vol3_Iss2_P7_498to505.pdf [Accessed on 23, July 2016]

Jiawei Han, Micheline Kamber, J. P. (2012). *Data Mining Concepts and Techniques*. Waltham: 243-267

Kaghyan S. & Sarukhanyan H., (2012). Activity recognition using K-nearest neighbor algorithm on smartphone with Tri-axial accelerometer. *International Journal of Informatics Models and Analysis*, *1*, 146–156. Retrieved from http://www.foibg.com/ijima/vol01/ijima01-2-p06.pdf

Kohavi, R. (1995). The power of decision tables. *Machine Learning: ECML-95*, 174–189.

Kwapisz, J., Weiss, G., & Moore, S. (2011). Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations …*, *12*(2), 74–82.

Ma, T.-Y., Lin, C.-Y., Hsu, S.-W., Hu, C.-W., & Hou, T.-W. (2012). Automatic brightness control of the handheld device display with low illumination. *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, 382–385.

Ma, Y., Xu, B., Bai, Y., Sun, G., & Zhu, R. (2012). Daily mood assessment based on mobile phone sensing. *Proceedings - BSN 2012: 9th International Workshop on Wearable and Implantable Body Sensor Networks*, 142–147.

Mailloux, J. (2013). Automatic Screen and Keypad Brightness Adjustment on a Mobile Handheld Electronic Device, *2*(12).

Maloof, M. A. (2002). On Machine Learning ROC Analysis and Statistical Tests of Significance, 204–207.

Marhoubi, A. H., Saravi, S., & Edirisinghe, E. A. (2015). The Application of Machine Learning in Multi Sensor Data Fusion for Activity Recognition in Mobile Device Space. *Image Sensing Technologies: Materials, Devices, Systems, and Applications Ii*, *9481*, 1–9.

Marhoubi, A. H., Saravi, S., Edirisinghe, E. A., & Bez, H. H. (2015). Illumination Modelling of a Mobile Device Environment for Effective use in Driving Mobile Apps, *9481*, 1–11.

Na, N., Jang, J., & Suk, H.-J. (2014). Dynamics of backlight luminance for using smartphone in dark environment, *9014*, 90140I.

Nagpal, V. (2016). *Android Sensor Programming by Example*. Birmingham:p 45-70

Nicolás Bourbaki, & Bourbaki, N. (1994). *Elements of the History of Mathematics*. Springer.p3-4

O'Nolan John. (2012). Rules for Perfect Lighting: Understanding The Inverse-Square Law. http://photography.tutsplus.com/articles/rules-for-perfect-lighting-understanding-the-inverse-square-law--photo-3483 [Accessed on 25, Aug 2012]

Panda, M., & Patra, M. R. (2008). A Comparative Study of Data Mining Algorithms

for Network Intrusion Detection. *First International Conference on Emerging Trends in Engineering and Technology*, 504–507.

Patil, T. R. (2013). Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. *International Journal Of Computer Science And Applications, 6*(2), 256–261.

Pearce, J. H. (1944). Fourier Series. *Nature*, *154*(3900), 130–131.

Perdo Domingos. (1997). Why Does Bagging Work? A Bayesian Account and its Implications. *AAAI*, 2–5.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. *San Francisco:p3-4*

Semiconductors, V. (2012). Ambient Light Sensors - Circuit and Window Design Mounting Vishay Semiconductors, 1–9.

Sensor, I. O. (2009). Intelligent OPTO Sensor Designer's notebook, (figure 1), 1–9.

Android.(2014). Sensors Overview Android Developers. http://developer.android.com/guide/topics/sensors/sensors_overview.html [Accessed on 24, Nov 2014]

Sharma, A. K. (2011). A Comparative Study of Classification Algorithms for Spam Email Data Analysis. *International Journal on Computer Science and Engineering (IJCSE)*, *3*(May), 1890–1895.

Sinusoid. (2012). *Encyclopedia of Mathematics*. Retrieved from http://www.encyclopediaofmath.org/index.php/Sinusoid

Smith, S. W. (1991). Moving Average Filters. In *Economics Letters* .Vol. 37, pp. 277–284.

SparkNotes: SAT Subject Test: Math Level 2: Identifying the Graphs of Polynomial Functions. (2011). http://www.sparknotes.com/testprep/books/sat2/math2c/chapter10section7.rhtml [Accessed on 12, Jan 2015]

Sunil Ray. (2015). 6 Easy Steps to Learn Naive Bayes Algorithm (with code in Python). https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/ [Accessed on 4, Sep 2015]

Thammasat, E. (2013). The statistical recognition of walking, jogging, and running using smartphone accelerometers. *BMEiCON 2013 - 6th Biomedical Engineering International Conference*, (1), 13–16.

Ustev, Y., Incel, O. D., & Ersoy, C. (2013). User, device and orientation

independent human activity recognition on mobile phones: challenges and a proposal., 1427–1435.

Waikato, University. Weka Data Mining Software in Java., http://www.cs.waikato.ac.nz/~ml/weka/ [Accessed on 18, Mar 2015]

Wikipedia. (2014). Six degrees of freedom - Wikipedia. http://en.wikipedia.org/wiki/Six_degrees_of_freedom#Robotics [Accessed on 23, Feb 2014]

Yang, H. C., Li, Y. C., Liu, Z. Y., & Qiu, J. (2014). HARLIB: A human activity recognition library on Android. *International Computer Conference on Wavelet Active Media Technology and Information Processing*, 313–315.