Dynamic Screen Based Lighting for Facial Detection

Ryan B. Griffiths

Supervisor: Dr Donald G. Dansereau

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Mechatronic Engineering (Honours)

School of Aerospace, Mechanical
and Mechatronic Engineering
Faculty of Engineering

The University of Sydney, Australia

November, 2020

Abstract

As phones have become relied upon within society to store sensitive information and facial detection has become more prominent. Having accurate facial detection methods on mobile phones is crucial. Because most phones only have a single camera to use for facial recognition, many facial detection systems can be fooled with a 2D image of a face, instead of the real thing.

This thesis hypothesises that the performance of these signal sensor phones can be improved using a novel approach which builds upon techniques developed in photometric stereo. The approach uses the screen of the mobile device to act as a varying light source, where multiple images can be taken and combined together to obtain information about the surface normal values that make up a face. One major goal for this approach is that it can work on a large variety of devices.

A laptop and mobile application were developed to test this hypothesis, which captures images and performs image processing, then feeds the image to a pre-trained classifier to determine real from fake faces. The app was tested in a range of environments, changing ambient light, number of images captured and movement of the imaged face. Correct classification of fake faces was demonstrated in situations, in which, the current method deployed on phones fails. In other situations, such as high ambient light, the proposed system performed badly.

The proposed method aims to work in conjunction with the existing methods, which are currently in use. Adding another layer of security to to the facial detection systems will incur an increase in processing time but will also provide increased performance and security for these systems.

Statement of Contribution

- I carried out the literature review on dynamic light techniques as well as mobile phone facial security; with guidance from my supervisor.

- I developed a MATLAB program to acquire images with different lighting, using the approach outlined in this work.

- I developed a mobile application to acquire images with different lighting & perform image processing on them, using the approach outlined in this work.

- With the help of my supervisor, I developed image processing algorithms to extract the required information from the captured images.

- I performed experiments & analysis on the developed system to evaluate its validity.

- I carried out the discussion and conclusion of the work achieved.

Ryan B. Griffiths
20th November, 2020

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr Donald G. Dansereau. He has been an invaluable source of information and advice throughout the whole process, whilst motivating me to achieve my best work. I would also like to thank him for the introduction to the topic investigated in this thesis along with the wider world of Computer Vision and research; in which, I have found a lot of joy and fulfilment.

I would also like to thank the Robotic Imaging Group. Our weekly meetings provided me with a great opportunity to display work and milestones achieved whilst also learning from others' experience.

Lastly, I would like to thank my friends and family who have supported me throughout my thesis and whole university experience which has led me to being where I am.

Contents

List of Figures

List of Tables

Introduction 1

Over the years computer vision has developed many different techniques for gathering more usable information about our environment using a variety of cameras. This additional information has lead to a variety of new applications being possible. This thesis outlines the process taken to investigate and build a portable dynamic screen based image capturing system, where images of the user's face can be taken under multiple different lighting conditions, leading to accurate facial detection with a single camera sensor.

1.1 Motivation

Mobile phones have become a pivotal part of everyone's day to day lives, being utilised for an ever increasing number of applications, such as emails, navigation, banking and appointments. These applications store sensitive information about the owner's location, bank details, future plans and passwords. If someone gains unsolicited access to a phone they would be privy to all of this information. The three typical methods for security in an Android mobile, are either a passcode, fingerprint, or facial recognition. Out of these three methods, facial recognition is the least secure [1], regardless almost every newly released phone offers it.

Virtually all Android phones only have a single camera sensor available to perform this facial recognition. The current method in which these phones provide facial detection is unreliable and can often be easily fooled by a single image of the user's face.

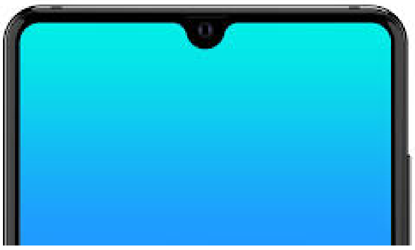


Figure 1.1: Android phones have a single camera sensor, which produces an unreliable result[1].

Phones that use multiple sensors, like that of the iPhone, in Figure 1.2, have an accuracy which is drastically improved, compared to the single sensor method, as it uses depth information to determine real from fake faces.

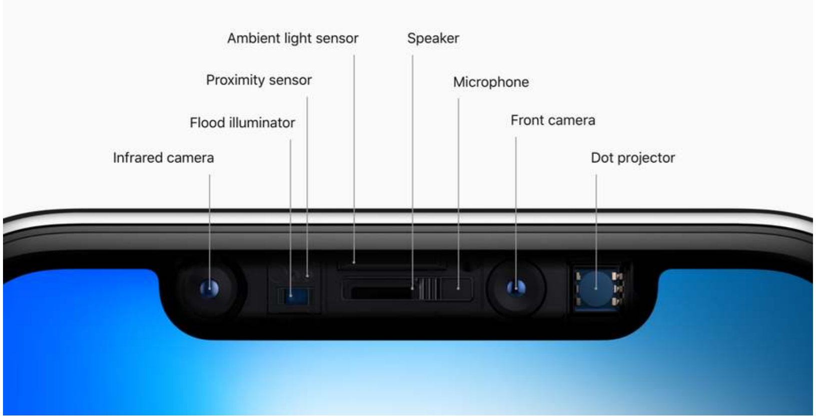


Figure 1.2: There are a multitude of sensors on the iPhone, providing high accuracy. [2].

Every Android phone has just a single sensor; except the Pixel 4, which has an array of sensors similar to that of the iPhone. However, this was a short lived addition to the pixel phones, as the new Pixel 5 removed these additional sensors, and went back to a single camera. The industry is heading in the direction of using facial recognition as the leading method of device security but is not adding additional sensors or reliability to its facial recognition. This increases the motivation of having a robust method of facial detection using a single sensor.

[1] Image sourced from KitGuru at https://www.kitguru.net/lifestyle/mobile/android/damien-cox/

[2] Image sourced from The Verge at https://www.theverge.com/2017/9/14/16306244/apple-iphone-x-design-notch

1.2 Problem Statement

The primary question being asked is whether it is possible to have a single camera phone produce results which are comparable to that of a multi sensor phone. The method in which this was investigated involves using dynamic lighting and taking multiple images, which are the principles behind photometric stereo. In this case, the lighting is provided by changing what the screen displays. this leads to some additional questions, such as: ′can using a screen as a light source produce adequate results to develop reliable facial detection?′ and ′how will it perform on a portable device that needs to work in a variety of environments?′.

The implementation of this investigation will be on a commercially available screened mobile device, such as a smart phone or laptop. The goal for this implementation is that it will work on readily available devices without any modifications. A mobile application was developed for this project, that takes images to recognise the user's face.

1.3 Contributions

This thesis extends on the work which has been performed in the photometric stereo, by expanding its use into uncontrolled environments. The following contributions have been made:

- Using a screen as a light source for photometric stereo.
- Ambient light removal, as the system is to work in as many situations as possible.
- As multiple images are being taken from a handheld device of a non static object (a face), it means that the images will have to be aligned to each other; otherwise the image processing will not be able to perform correctly.
- Image processing required to extract the relevant information from the images, with unknown light source

values. The required information will be an indication of the surface normal values.

1.4 Thesis Outline

Chapter 2 - provides a background on the general principles and methods that have been utilised within the thesis, such as the image acquisition, dynamic lighting and classifiers.

Chapter 3 - is a review of literature within this space; looking at light stage and photometric stereo designs, facial recognition techniques and different spoofing methods and preventions which have already been developed.

Chapter 4 - follows the method used throughout the development process, showing both the processes from the laptop and mobile environments. Also looking at the how the results are obtained.

Chapter 5 - shows the results gathered under differing conditions and any insights gained from them that might be useful moving forward.

Chapter 6 - discusses the system, how well it performed and fundamental limitations that using screen based lighting has. This chapter also includes a comparison to existing solutions.

Chapter 7 - concludes the system and provides some details on future work to be done.

Background 2

This chapter gives some brief background information into the concepts used throughout this process.

2.1 Image Acquisition

The goal in this section is to give a brief background into how images from a camera are obtained. This is achieved by looking at the mathematical representation of pixel intensity values and the noise that the image contains. The albedo is a term used in imaging to denote the reflectivity of the object in the scene; essentially, how much light it reflects back off its surface. Albedo and reflectivity have been used interchangeably within this thesis.

2.1.1 Rendering Equation

The numerical model for image acquisition can be described using the rendering equation. This equation describes the light seen from an observer in a specified position, in this case, for this thesis, the observer is a camera sensor [2]. The equation takes into account many different factors. A Graphical representation is shown below in Figure 2.1.

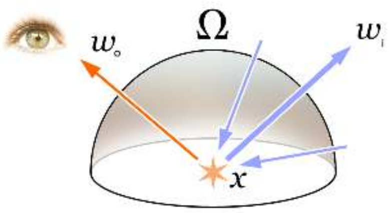
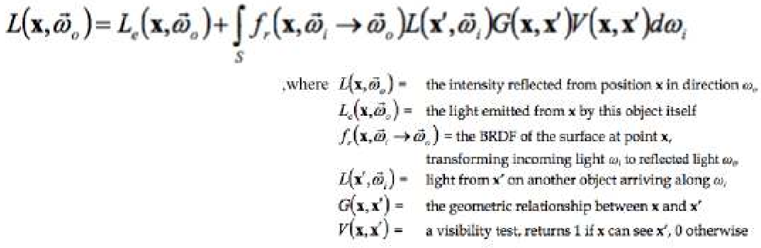


Figure 2.1: Graphical representation of Rendering Equation [1].

[1] Image sourced from Wikipedia at https://en.wikipedia.org/wiki/Rendering_equation

The full rendering equation has many terms, meaning it can be difficult to model and interpret. See Figure 2.2 for the full equation.



$$L(\mathbf{x},\vec{\omega}_o)=L_e(\mathbf{x},\vec{\omega}_o)+\int_S f_r(\mathbf{x},\vec{\omega}_i\rightarrow\vec{\omega}_o)L(\mathbf{x}',\vec{\omega}_i)G(\mathbf{x},\mathbf{x}')V(\mathbf{x},\mathbf{x}')d\omega_i$$

,where $L(\mathbf{x},\vec{\omega}_o)$ = the intensity reflected from position **x** in direction ω_o
$L_e(\mathbf{x},\vec{\omega}_o)$ = the light emitted from **x** by this object itself
$f_r(\mathbf{x},\vec{\omega}_i\rightarrow\vec{\omega}_o)$ = the BRDF of the surface at point **x**,
transforming incoming light ω_i to reflected light ω_o
$L(\mathbf{x}',\vec{\omega}_i)$ = light from **x**′ on another object arriving along ω_i
$G(\mathbf{x},\mathbf{x}')$ = the geometric relationship between **x** and **x**′
$V(\mathbf{x},\mathbf{x}')$ = a visibility test, returns 1 if **x** can see **x**′, 0 otherwise

Figure 2.2: Full Rendering Equation

The analysis done in this thesis uses a simplified version of this equation in which many of the terms have been grouped together. This has allowed for easier understanding of how best to process the images in order to extract the desired information. In Equation 2.1, it shows the simplified version.

$$Y = \sum(L_i R(V_i \cdot N)) \tag{2.1}$$

where the L_i is the intensity of the light source, R is the reflectively of the surface, V_i is the direction of the light source from the surface, N is the surface normal to the camera and Y is the intensity of the pixel recorded by the camera.

2.1.2 Image Noise

Noise within images is something that has to be considered in every application. It is the result of physical characteristics of how camera sensors capture the emitted photons within a scene. There are different sources of noise which all contribute to the overall image [3].

The majority of noise types are signal independent, meaning that they provide the same effect regardless of what is being imaged. These include gaussian noise (see Figure 2.3) which is largely due to thermal noise.

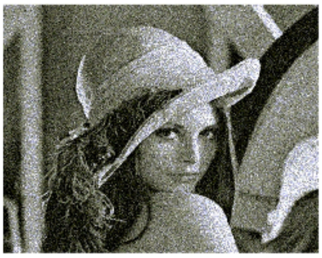


Figure 2.3: Image with Gaussian noise[2].

Salt and pepper noise, see Figure 2.4, is due to the errors within the digital representation of the analogue signal. Photo or shot noise on the other hand changes with the increases, proportionally to the square root, of intensity.

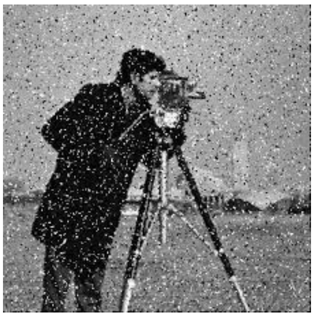


Figure 2.4: Image that has salt and pepper noise present[4]

This means, in most cases, imaging brighter objects leads to a higher Signal to Noise Ratio (SNR), as the noise stays largely the same; with the exception of photon noise which increases with the square root of intensity, whilst the signal increases proportionally to the intensity.

2.2 Imaging with Dynamic Lighting

2.2.1 Photometric Stereo

Photometric stereo is a method of image capturing which is used to gather surface normal and depth information when imaging. This is achieved by taking multiple images under

[2] Image sourced from Research Gate at `https://www.researchgate.net/figure/Noisy-image-Gaussian-noise-with-mean-and-variance-0005_fig2_252066070`

different lighting from specific and known locations. Using the information of the light source direction, as well as the captured image brightness, the surface normal values can be obtained as they are a function of the image brightness and light source locations. A minimum of three images under different lighting conditions are required to solve the equation simultaneously. Woodham et al. [5] first utilised this method back in the 1980.

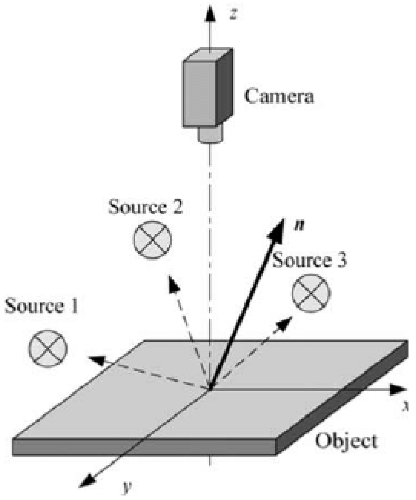


Figure 2.5: Different light sources illuminate the subject, to produce different illumination patterns[3]

Photometric stereo has been well established and researched using these conventional methods, with the results that are achieved being of a high standard.

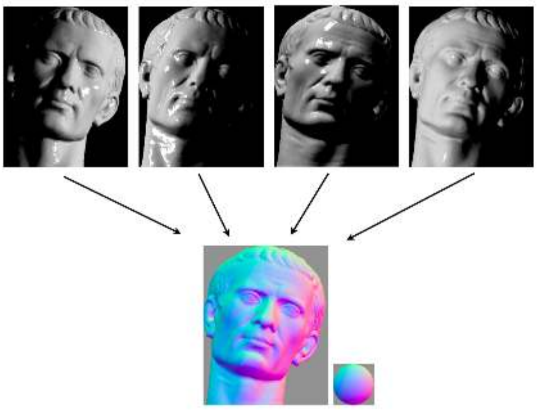


Figure 2.6: Example of photoelectric stereo for obtaining facial surface normals[4]

[3] Image sourced from Research Gate at https://www.researchgate.net/figure/Principle-of-photometric-stereo_fig7_222422584

[4] Image sourced from Wikipedia at https://en.wikipedia.org/wiki/Photometric_stereo

2.2.2 Light Stages

A light stage is an apparatus in which the lighting of a subject area is varied is a systematic way, this enables images to be captured of the scene with more information. This is achieved using the same principles as photometric stereo, but take it a step further. High detail can be obtained from the scene's textures, albeo values, and surface normals [6]. A traditional version of a light stage would be to have a constellation of fixed lights and cameras surround the area in focus, as seen in Figure 2.7.

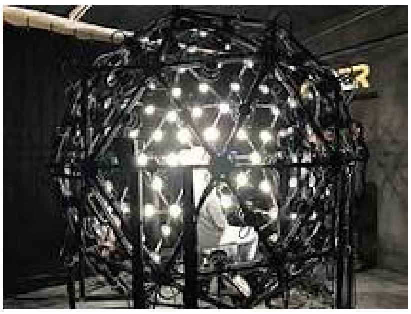


Figure 2.7: Existing Light Stage apparatus[5]

The light stage will capture images under a range of different lighting patterns.

2.3 Image Registration & Ambient Light

2.3.1 Aligning Images

Aligning images is a crucial step in many real world applications in computer vision. It is an important step in many medical imaging processes as they are required to combine multiple images together [7]. Two versions of image registration are rigid and non rigid. Rigid alignment only performs a transformation that looks at the image as a whole. There are different transformation types which can be performed, such

[5] Image sourced from Wikipedia at `https://en.wikipedia.org/wiki/Light_stage`

as translation, scaling rotation or projective. The method in which this is completed can either be intensity matching or feature matching [8]. Where as, the non rigid method provides a move complete alignment in which the objects in the image are no longer rigid but plastic. This is particularly import when imaging dynamic objects. One non-rigid method is Demon's algorithm. This method is an iterative process in which the images converge to be aligned to one another[9].

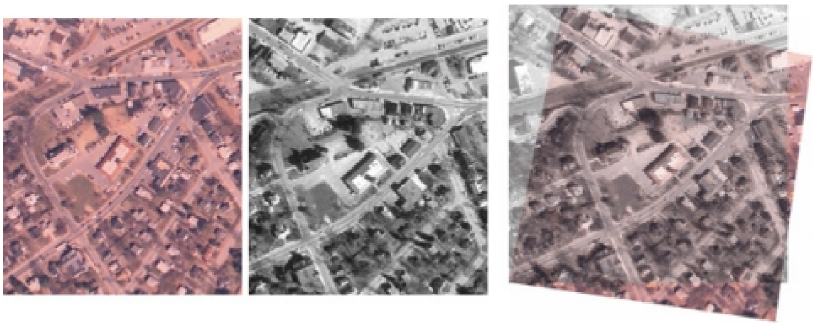


Figure 2.8: An example of rigid registration of two images[6]

2.3.2 Removing Ambient Light from Images

In certain situations, such as this thesis, it is a requirement to find the illumination in an image due to a specific light source while there is ambient light present. The process required to remove this ambient light from a captured image is relatively simple, although it requires having multiple images of scene, with and without the additional light source. Additionally, the ambient and illuminated images are required to be aligned with each other, hence, this process works better in static scenes. Comparing the illuminated image with the ambient light image, through subtraction of the ambient light image from the illuminated image, provides an image with just the illumination of the object due to the additional light source [10].

This has been expressed mathematically below using the simplified version of the rendering equation that was discussed

[6] Image sourced from MathWorks at `https://au.mathworks.com/discovery/image-registration.html`

earlier.

$$Y = \sum(L_i R(V_i \cdot N)) \tag{2.2}$$

The L_i is the intensity of the light source, R is the reflectively of the surface, V_i is the direction of the light source from the surface, N is the surface normal to the camera and Y is the intensity of the pixel recorded by the camera.

Now, if Y_0 represents the ambient light image, all the ambient light sources can be grouped together to form one term.

$$Y_0 = L_0 R(V_0 \cdot N) \tag{2.3}$$

Taking another image with the additional light source, means that there is an extra light source present to add to the equation.

$$Y_1 = L_0 R(V_0 \cdot N) + L_1 R(V_1 \cdot N) \tag{2.4}$$

If image Y_0 is subtracted from image Y_1 only the light added by the additional light source is left.

$$Y_1 - Y_0 = L_0 R(V_0 \cdot N) + L_1 R(V_1 \cdot N) - L_0 R(V_0 \cdot N) = L_1 R(V_1 \cdot N) \tag{2.5}$$

2.4 Classifiers & Facial Detection

2.4.1 Classifier Techniques

Classifiers are an integral part of many imaging systems. In relation to this thesis, the classifier is what performs the final output of what the image is of. In this case, it will be to recognise if the image is of the correct live face. The way in which the classifier works is in two stages. Firstly, it breaks up the image/input given to it into a list of different features and the occurrence of these features. Based on these features the input is then classified by comparing it to a predetermined criteria. The criteria is developed during the training stage of the classifier, where it learns what list of features represents each classification. Figure 2.9 demonstrates how an SMV (support vector machine) would group images based of the vector of features [11].

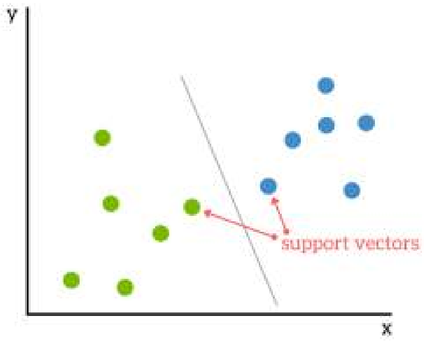


Figure 2.9: Support Vector Machine Example[7].

Feature extraction is an important step within image classification. A bag of Words, also called a bag of visual words, is a concept which is used to extract groups of features from the image [13]. It treats features in the images as words. These words are counted and stored in a vector. Based on the occurrence of these words an image can be classified. This allows for a reduction in feature space as they are grouped together as words and counted. This method still requires a classifier to be utilised to decide what bag of words belongs to which category; such as an SVM or some other approach.

Neural networks are a slightly different approach to achieve the same goal. They are vaguely based off the neurons within a brain, hence the name. A network is made up of these artificial neurons, which are then trained using a database, much like the conventional machine learning classifier [16]. The training process adjusts the parameters on the neurons within the network so that it responds in the correct way to classify the different categories. Figure 2.10 shows an example a typical network construction.

[7] Image sourced from Kdnuggets at `https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html`

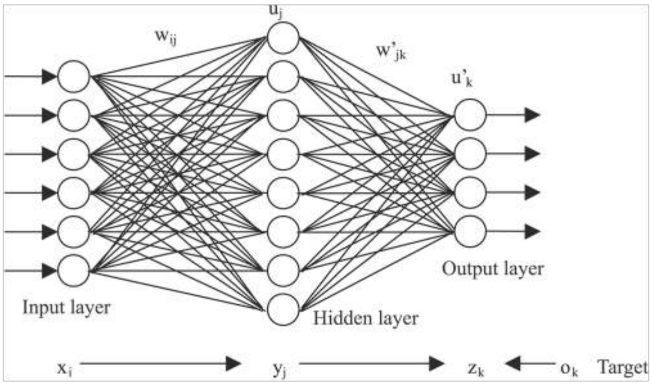


Figure 2.10: Example of a neural network[8].

2.4.2 Facial Recognition

Facial recognition techniques have been well explored, as it is an expanding industry with many applications, from security to marketing. There are many well established systems and principals behind the recognition. One general principal behind the process is to firstly identify key features within a face. These could be features such as eyes, eyebrows, lips, et cetera (see Figure 2.11). The method in which these features are identified is through a classifier, which could use a neural network or a bag of features and SVM, [18].

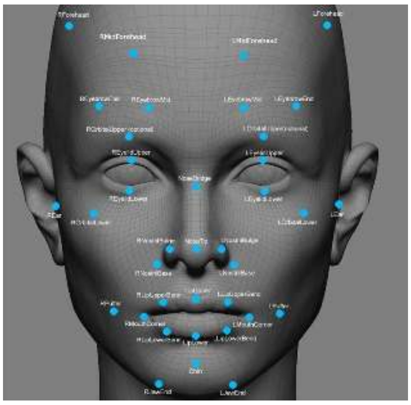


Figure 2.11: Features used in existing facial recognition techniques[9].

Once the feature's position and orientation are found, the geometry of each of the features can be obtained and compared to reference faces to see if they match.

[8] Image sourced from ExtremeTech at https://www.extremetech.com/extreme/215170-artificial-neural-networks-are-changing-the-world-what-are-they

[9] Image sourced from Learning Hub at https://learn.g2.com/facial-recognition

Facial detection is different to recognition. This project attempts to built a facial detection algorithm which can be utilised in preexisting facial recognition systems.

Literature Review 3

This chapter details the literature that has been reviewed to gain insight into current methods surrounding the topics investigated in this thesis, including light stages, photometric stereo, screen based light sources and facial recognition spoofing.

3.1 Light Stages

The concept of light stages was pioneered by Paul Debevec. His method uses a constellation of RGB lights, which illuminate the subject from all angles and mimics lighting environments. This method builds on from photometric stereo, with the goal to produce higher performance. Debevec achieved extremely high results from his methods which have been widely used, from applications such as blockbuster films simulating realistic lighting conditions, to generating a highly detailed model of Obama's head [19]. The applications in which these light stages are used often want to find the reflectively of the objects [6], where in this thesis the reflectively needs to be removed from the captured images. There are limitations with this process though: the extensive setup costs and time, which exclude many of the potential users who would benefit from this solution; and poor portability and availability of this method.

Effort has been made in the past to develop a portable light stage, demonstrated by the University of Leuven. [20]. Their method was to have a mounted camera with a single light source which was moved around the subject by hand for many orientations. More recently, in 2017, an attempt was made to produce a portable light stage, mounting lights and cameras on drones and using them to produces a constellation of lights [21]. Both of these methods differ substantially from the approach in this thessis to build a portable solution.

3.1.1 Classification Using Light Stages

Image classification is a category within the light stage space which has a lot of promise as it can utilise the additional information captured to make better educated designs for classifications. Using light stages for classification is something that has not been used in a wide range of applications. The choice of which lighting patterns to use to give the best results is not obvious. Schechner et al. [23], explores how different lighting patterns and multiplexing can be used to maximise the SNR, which would in turn produce better classification results. Wang et al. [24], explored explicitly the process of what lighting patterns produce the best results for classification. This thesis does not investigate this process in depth, but it would be an important consideration for improving the results further.

3.2 Photometric Stereo

As stated, photometric stereo was first developed in the 1980s; as such, it has been thoroughly explored within the literature for a range of different applications and use cases. Zafeiriou et al. [25] and Zhou et al.[26] explored using photometric stereo for the application of facial recognition, by building a data set of face images which can be used for classifier training. The approach taken used a more conventional method, which was explained in Section 2.2.1.

3.2.1 Unknown Light Sources

Initially, the requirement for photometric stereo applications was that each of the light sources had to be known because this information was used in the calculation of the surface normal values. The idea that its possible to use unknown light sources has been investigated. Woodham et al. [27] in 1991 and Miyazaki et al. [28] in 2010 have both shown that this is possible. Over time these approaches estimate the locations of the light sources.

The method, which is utilised in this thesis, also has unknown light sources, as it is generated by a screen. The way in which this was dealt with is a different approach to the methods within the literature and is discussed in detail further on.

3.2.2 Point Light Sources

Photometric stereo, like many things, usually deals with light sources as a point. Essentially, it is assumed the the light source is emanating from a single point in space, where it has width to the light source. This assumption often doesn't hinder the results too much because the sources are fairly small and far enough away. Mecca et al. [29] looks at what occurs when light sources are brought close to the object being imaged. The assumption that the light source is a single point starts to breakdown.

This is a consideration which this thesis needs to account for. Not only is the light source placed extremely close to the subject, the actual light source is from a screen, where the light is being generated by a full panel, with a significant surface area.

3.3 Screen Based Light Sources

Using a screen as a the light source for imaging is something which has not been developed or validated extensively. This concept is the pivotal part of this thesis. Throughout the literature there are a few papers which investigate this. Funk et al. [31] in 2007, looked into this, where they were able to produce surface normal values, for the scene being imaged, by using this technique. See Figure 3.1 for the experimental setup used.

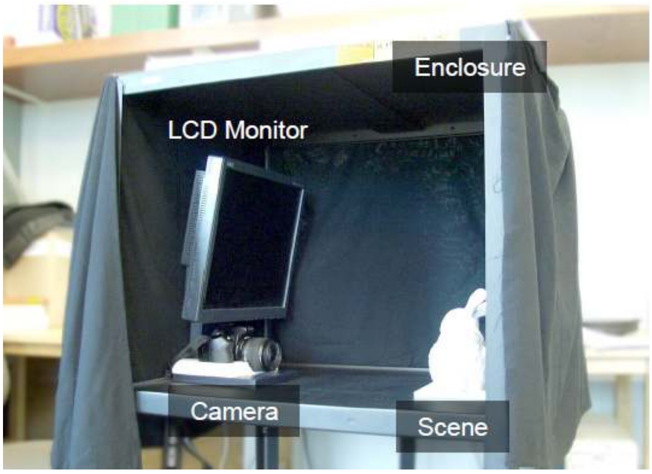


Figure 3.1: Image of previous screen based approach, experimental approach. Funk et al. [31]

Bi et al. [32] in 2014, produced a 3D reconstruction of a model head using this same approach. The results from this produce a good likeness of the model head. The results from this paper show a lot of promise for this method of photometric stereo.

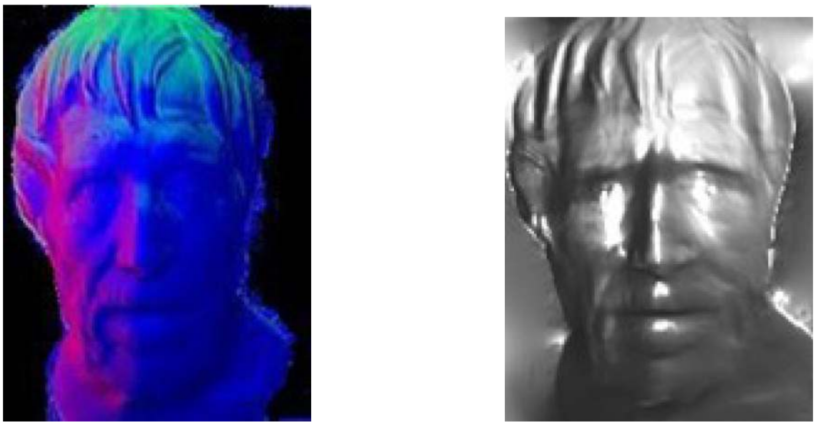


Figure 3.2: Image of previous screen based approach, 3D Reconstruction of model head. Bi et al. [32]

These papers show that using a screen as a light source for photometric stereo is a valid option. The approaches taken though still suffer from the same limitations as discussed in Section 3.1. Both systems operated purely in dark environments and only on static objects. Which limits the use cases dramatically. These approaches also require the location of each of the illuminations on the screen to be predetermined and calculated, in order to determine the surface normal values in the scene.

The other thing to note is that screen technology has developed rapidly over the years. The displays used by Funk. in 2007 and even Bi. in 2014, are considerably different to the average display available currently in 2020. The biggest development, that would benefit this application is OLED display. It can achieve greater contrast between lights and dark pixels, due to its single pixel illumination compared to a whole display back light.

3.4 Facial Detection

Facial detection is the first important step in facial recognition. The difference being, facial detection will detect any face within an image, while facial recognition will determine specific faces in the image, which obviously requires faces to be determined first. The performance of the facial recognition relies strongly on the facial detection performance. Jain et al. [33] looks in depth at the facial recognition techniques and facial detection. Facial detection within an image will perform some prepossessing, such as image normalisation. This is then passed to a classifier, which could be either neural networks or more conventional machine learning.

One important relationship that facial detection systems have is their false alarm rate vs their detection rate. Ideally 100% of real faces get seen as a face while 0% of everything else is seen as a face. This is never going to be the case, meaning that a compromise needs to be made between the these values. Below, Figure 3.3 shows the typical relationship between these values.

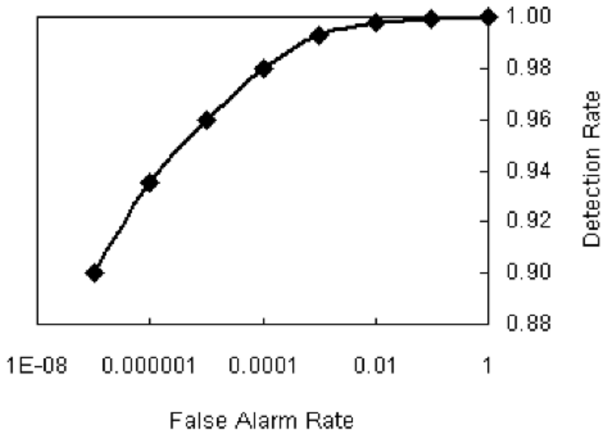


Figure 3.3: Typical relationship for facial detection false alarms vs detection rate. Jain et al. [33]

There are other methods of facial detection which use different sensors other then cameras. Pan et al. [34] uses hyperspectral images for facial detection. However, these approaches do not align with the goal of it being usable on commercially available devices.

3.4.1 Smart Phone Facial Security

As stated before, the application of facial recognition to unlock your phone, is done using one of two methods. One method is conducted using a single camera sensor; which takes a photo and compares it to the reference, making sure they match. This method is most common in Android phones [35]; where significant research has been done to try and detect live faces from static images, such as motion detection or eye blinking[36]. The second method, commonly used on Apple phones and the Google Pixel 4, involves using a multitude of sensors. Both of these devices use infer red (IR) cameras along with dot projectors to develop depth information and structure of the face being observed[37]. This allows them to build comprehensive models. The other sensor used is a flood illuminator, which allows the device to be unlocked, even in the dark. This method is vastly more secure then the first method. However, it requires multiple sensors and takes longer amount of time [38].

By using a new approach, there is potential to create images with much more information. This method will likely not attain the same high results that the industry has achieved but there is potential to achieve great results. Because there will be much more information gathered, this extra data can be used to perform high level facial recognition, using just a single sensor. Thus, providing greater security when using facial recognition to unlock a device.

3.5 Spoofing Techniques and Detection

Spoof detecting is becoming an important process to be able to do for many applications. As biometrics become more regularly used, it requires more attention. The topic has been explored in a manner of ways. One paper, "Spoofing in 2D Face Recognition with 3D Masks and Anti-spoofing with Kinect" by Nesli Erdogmus and Sebastien Marcel [39], details their approach to this topic. They produced 3 dimensional marks based on images of faces. These masks were then used to spoof a 2D facial recognition system. The method in which they attempted anti spoofing was in 2 separate ways. Firstly, using the surface texture of the faces, compared to the masks, as there will always be some texture difference between the two. The second method uses a depth camera found on an Xbox Kinect. The information gathered from this depth camera is then used to determine differences in shape from the true face and the masked version. Better results were attained from using the depth camera compared to texture analysis. Additional hardware was required to archive better results.

Another paper explores the use of eye blinking detection as an anti-spoofing method [40]. This approach moves away from the classification based around shape and texture,instead opting for testing the ’liveliness’ of the face being imaged. There are other features to measure besides blinking, such as facial expressions or mouth and head movements [40]. This method worked well; although, on its own it has no way to determine the shape or surface being imaged. Meaning if a system used this method it would have to rely on other options to protect against an attack. An attack could include a video being played of a face instead.

The methodology for this thesis takes a different look at how to categorise a spoof in the scenario of a 2D version of a face. Instead of using additional camera hardware to build the structure of the face being imaged, dynamic lighting is used instead. The thought is that this classification on surface normal values will be able to perform as well. Another point to make, is that light stages have the ability to produce much better detail on the texture of the surface being images, meaning it could be possible to perform classification centred around texture instead of surface normals. This has the potential to produce better results, though it has not been explored within this thesis.

3.6 Summary

The current methods for using dynamic lighting to determine surface normal values of a scene can produce great results. However, they are only performed in dark rooms and on static objects. They are also not readily available or portable solutions; which are some of the main goals of this thesis.

The approach in this thesis differs conventional light stages and photometric stereo because it has been completed using ready-made commercial screened based products as the lighting apparatuses. It was taken further then the other screen based lighting solutions which have been developed, as it is required to work in a large variety of environments. For this to occur, there are two main contributions that need to be made. Firstly, it is important that the ambient light is accounted for and removed within each of the images. Secondly, the non static nature needs to be accounted for.

The other consideration to make is that it is not possible to know the light source locations. Also, as stated, the light source can not be considered a single point. This means the conventional approach of calculating the surface normal values directly is not possible. Luckily, for the classification of a face by its shape, it is not a requirement to know the exact values of the surface normal values. Instead the values found can be a function of the light source direction, intensity and surface normals. These light sources are unknown but can be kept consistent across all images, meaning it would effect the results. See Section 4.2.3 for a full method.

The current facial detection methods deployed on a single camera mobile phone is not reliable. The goal in this thesis is to improve the results by applying a new approach which has not been explored within the literature. The real question that this thesis pivots on, is if this mobile screen based light source can produce sufficient results for the system to be able to produce a reliable facial detection application. Research shows, there have been no other attempts of adapting the light stage concepts and methods for a mobile facial detection system.

Method 4

In this chapter the method followed so far is outlined, including the progression of development that has occurred along the way; looking at any complications that were encountered.

4.1 System Overview

The goal of this system is to extract an indication of the surface normal values from the captured face as accurately as possible. This information is then fed into a classifier, to tell if the surface normal values are consistent with a real or fake face. The system was prototyped in MATLAB, while an end to end system was developed as a mobile application to test and demonstrate its validity to work on a handheld device.

Here, a brief overview of the different modules that were developed is outlined. These different modules work together to condense the information in the captured images to extract the surface normal values. A diagram representing the flow of information and the different processing is shown in Figure 4.1

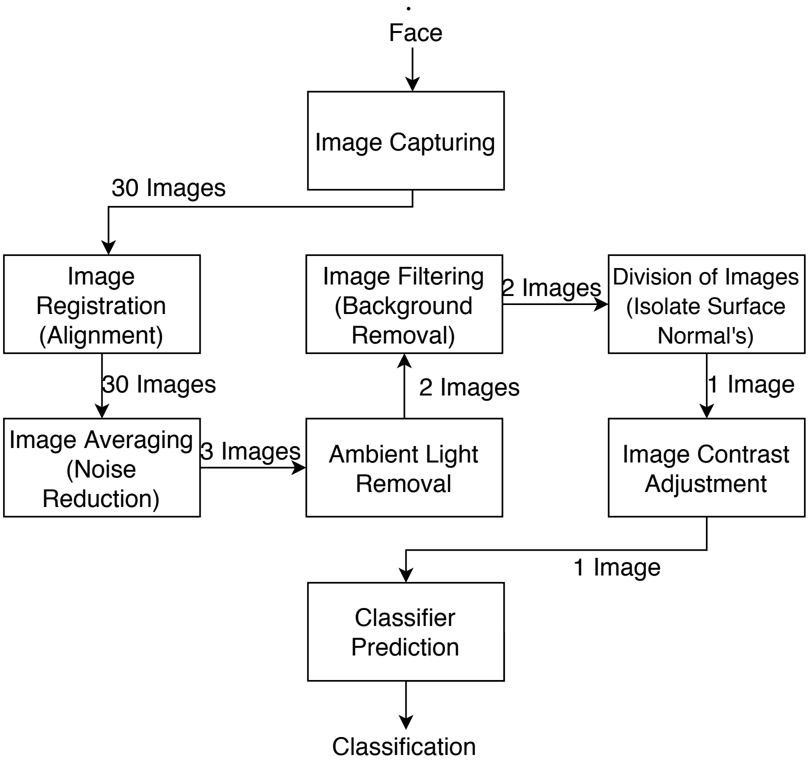

Face
Image Capturing
30 Images
Image Registration (Alignment)
30 Images
Image Averaging (Noise Reduction)
3 Images
Ambient Light Removal
2 Images
Image Filtering (Background Removal)
2 Images
Division of Images (Isolate Surface Normal's)
1 Image
Image Contrast Adjustment
1 Image
Classifier Prediction
Classification

Figure 4.1: Overview of the different modules, showing the number of images between each module. Showing the condensing of information that takes place.

The image capturing involves taking multiple images with 3 different lighting patterns. Below, in Figure 4.2, is an expanded view of the image capturing box in the previous image.

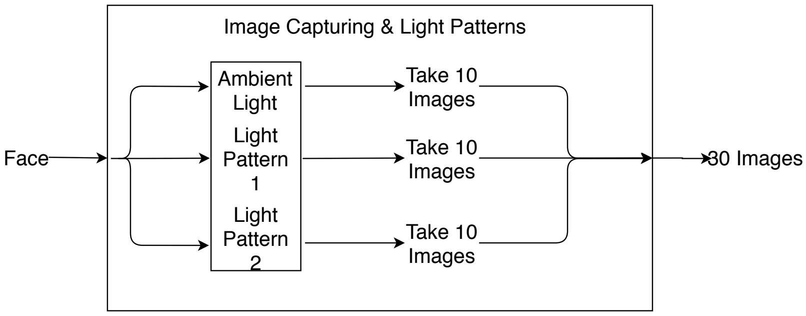


Figure 4.2: Overview of how the image capturing process combines groups of captured images.

4.2 Prototyping Laptop Environment

For easy development and options, the system has been implemented first on a computer platform in MATLAB. This is because MATLAB allows for rapid development, especially in regards to matrix manipulation, which is required for the image processing. This was done with the goal that the lessons learnt from this will reduce the time required for development when implementing the final system to a higher standard. The development process so far has been conducted in a computer environment using MATLAB. The implementation was done in the following order:

1. Screen-Based Light Stage
2. Image Acquisition
3. Image Processing
4. Image Classification

4.2.1 Screen-Based Light Source

The first step in the development process was to build a basic screen based light stage. This was completed in MATLAB by displaying a binary matrix, with values of 0 or 1, to the screen. This will be displayed on the screen as either a black or white pixel. A white pixel will give off more light then a black one. By toggling pixels between states a light stage was produced.

This was implemented in a way that would allow for the changing of certain parameters within the light stage. This included the size (number of pixels) that the light stage took up on the screen. The number of lighting sections within the stage was variable. It was important to be able to vary the number of images captured at each pattern. This allowed for investigation into the averaging to take place as a noise reduction technique (increasing the SNR).

Throughout the prototyping process the lighting pattern used was just vertical and horizontal strips of illuminated area, as can be seen in Figure 4.3.

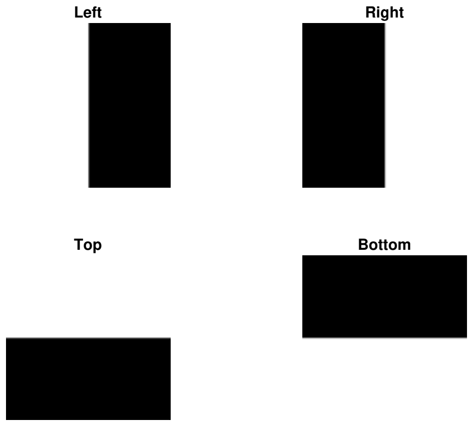


Figure 4.3: Image of different lighting patterns used in the laptop environment.

In preparation for the need to remove ambient light from the images taken, images were taken with a completely blank pattern. This will allow the ambient light to be subtracted in the image processing stage.

4.2.2 Image Gathering

The next step was to setup the webcam to run along side the light pattern. The images were taken as grey scale to reduce the complexity of processing required. Because the value of interest is the surface normal's, taking the grey scale does not remove this value from the image.

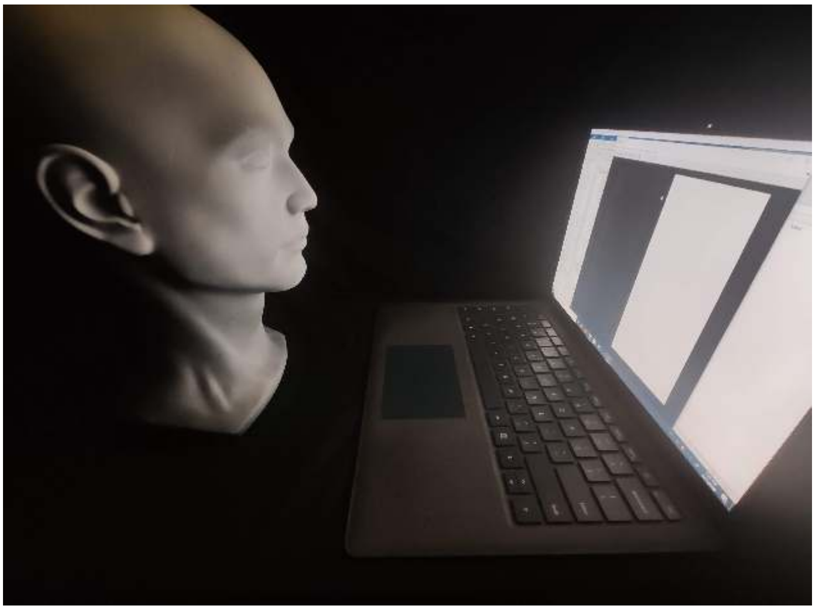


Figure 4.4: Image of Physical Setup used to capture images in the laptop environment.

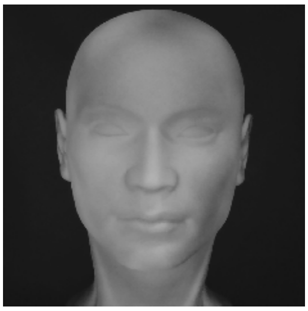


Figure 4.5: Grey Scale Image of Model Head

The first step is to take an ambient light image. The next step was to then take another image once the desired light pattern was displayed. Multiple images were taken for each pattern with an ambient light image taken between each one.

The synchronisation of when the lighting patterns show and the capturing of the image was necessary. This process required delays to be put in place, which lengthened the time taken to acquire all the required images. This process can definitely be refined to reduce the time taken but for the prototyping environment it is acceptable.

4.2.3 Image Processing

Image Registration

Image registration is required for this application because it will be implemented on a hand held device whilst also imaging a non static object, a face. The image processing performed after this step requires each image to be aligned to each other so they can be directly compared. If the images are misaligned then the small amount of signal within the images is completely overshadowed by the slight variances in the alignments.

Two separate methods were tried for the image alignment; rigid and non rigid transformations. As explained in Section 2.3.1, non rigid transformations can account for movements of plastic object. While imaging, a face will unlikely remain completely rigid, which means that a rigid transformation is never going to perfect result. The non rigid method, on the other-hand, might but it will also likely take longer (though this can be adjusted). Both the rigid and non rigid image registration was implemented using inbuilt MATLAB functions. The rigid transformation type was projective. The non rigid transformation algorithm was implemented following Demon's algorithm.

Below, in Figure 4.6, is an example of how each separate method performed. The algorithms were run on static images to start with to see how they perform.

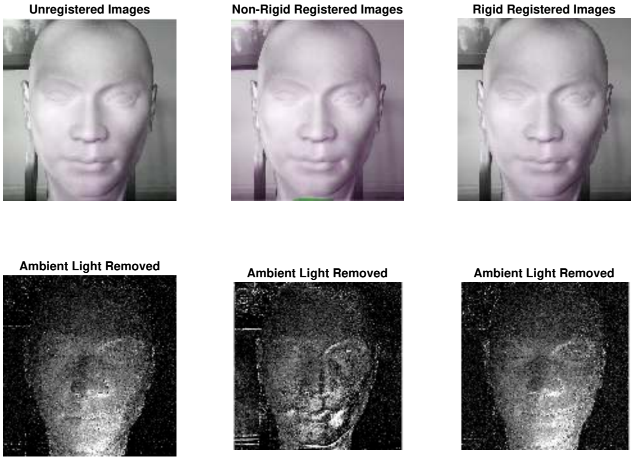


Figure 4.6: Comparison of rigid and non rigid image alignments algorithms, when there is no movement.

The same rigid and non-rigid algorithms were run again, now of a moving target.

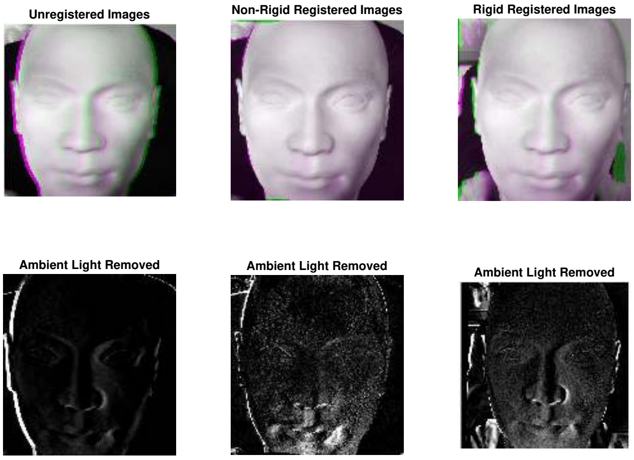


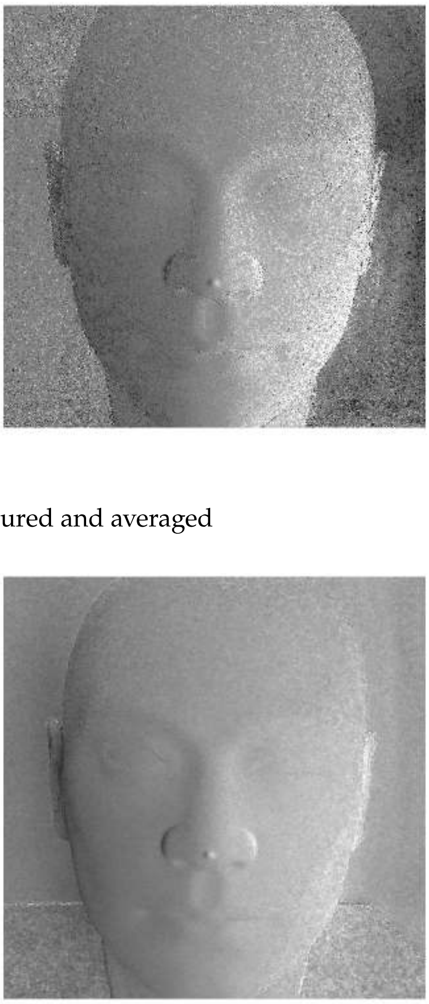
Figure 4.7: Comparison of rigid and non rigid image alignments algorithms, when there is movement between images.

As can be seen, the non rigid transformation actually degraded some of the desired light signals within the image with its transformation process. As the algorithm shifts the face around to try and match the images, any inaccuracies in the transformation, resulted in the light signal on the face to be moved along with it. The rigid transformation on the other hand performed better as the general alignment across the whole face was performed well.

Theoretically, non rigid registration would likely perform better, if the algorithm was tailored to task, with all the correct parameters chosen. Having said that, going forward rigid transformations were selected to be used. The reasoning behind this is that the rigid transformation, provides good results while it requires less optimisation and is potentially more reliable as it is a simpler process.

Averaging & Noise Reduction

An average is taken of all the images captured for each specific light pattern. This reduces the noise present because the noise is non biased. Averaging the values for each pixel across multiple images will result in a convergence of the value towards its true (noise free) value. The following images are actually a result of additional processing which is explained further, but as it shows a good example of the averaging results they are included here.



(a) 2 images captured and averaged

(b) 10 images captured and averaged

Figure 4.8: Images with different number averaged

The two figures shown in Figure 4.8, illustrates visually the effects of averaging images, and the noise reduction possible. This relationship is investigated further in Chapter 5, looking at how it effects the overall performance of the system.

Ambient Light Removal

It is required to remove ambient light from each of the captured light pattern images. This is done by subtracting the accompanying ambient light image from the light pattern image, as explained in Section 2.3.2. The mathematical representation for this has been restated below.

$$Y_1 - Y_0 = L_0 R(V_0 \cdot N) + L_1 R(V_1 \cdot N) - L_0 R(V_0 \cdot N) = L_1 R(V_1 \cdot N) \tag{4.1}$$

In this case, Y_0 is the ambient light image, Y_1 is each of the images with the additional light from the lighting pattern on the screen. This leaves an image with only the additional light present for each pattern used. Figure 4.9, shows the results from from this process.

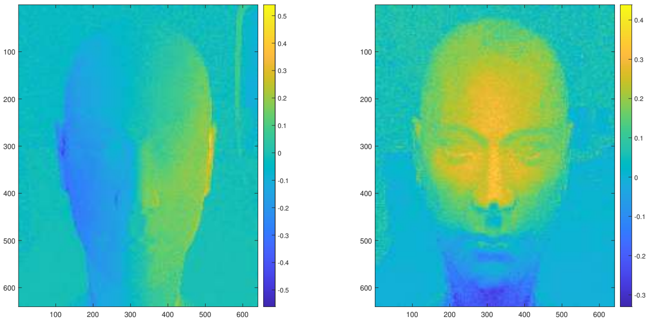


Figure 4.9: An example of ambient light being removed from the image.

Image Manipulation

The next step in the image processing method is to remove the albedo term from the image as seen in equation (2.2). Because this term is constant across all images, it is reasoned that it can be removed by the division of one light pattern from another.

$$\frac{Y_1 - Y_0}{Y_2 - Y_0} = \frac{L_1 R(V_1 \cdot N)}{L_2 R(V_2 \cdot N)} \tag{4.2}$$

This equation is then simplified to remove the albedo or reflectivity term R,

$$\frac{Y_1 - Y_0}{Y_2 - Y_0} = \frac{L_1 (V_1 \cdot N)}{L_2 (V_2 \cdot N)} \tag{4.3}$$

so that it now only contains L, V and N terms. Where L and V are both a result of the light stage and can be controlled to some effect. This means that the result idealistically would only vary with changes in the surface normal value N. With this result an acceptable representation of the shape of the subject being imaged can be achieved.

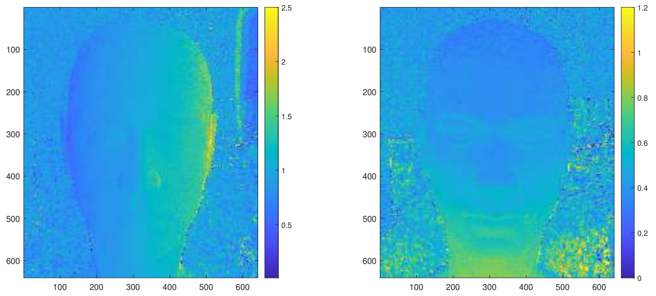


Figure 4.10: Left pattern divided from Right(left), Bottom pattern divided from Top (right)

With the nature of division, noise can unfortunately be amplified. Therefore, filtering should be implemented to extract out only the area that is of interest.

Filtering

This was achieved by checking the value of each pixel to see if its value across all of the lighting conditions was less than a specified threshold. This is essentially checking to see if the pixel in question is in the foreground or background of the image. The filtering implemented works well if the cutoff threshold value is adjusted for the different lighting conditions. Currently, the filtering has different effects depending on if the image set is taken in low ambient light or high ambient light. For this reason it is not universally reliable.

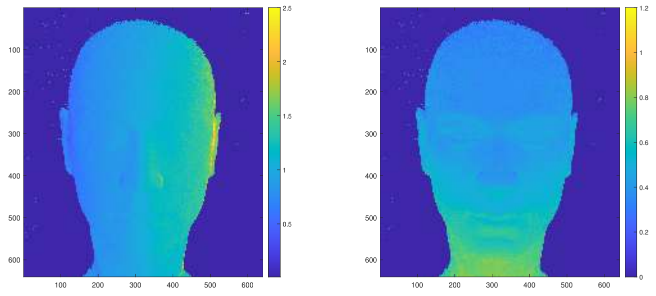


Figure 4.11: Filtering at different light levels

The mathematical representing for how the filtering occurs is shown.

$$Image_{Filtered} = (Pixels(Image_1 + Image_2) < Threshold) \tag{4.4}$$

Where $Image_1$ and $Image_2$ are the ambient light removed images for the different lighting patterns. The threshold value required depends strongly on the amount of ambient light present when imaging.

4.2.4 Classification

With the previous image processing done, the resulting images can be fed to a classifier. The classifier featured in this thesis was built using a MATLAB example for a bag of words. Prebuilt functions within MATLAB were used to train the classifier against a data set gathered. This classifier is a category classifier, meaning it will place the image it is fed into the most likely category it has been trained to recognise. There are only two categories trained, either it is a model head or not a model head.

The implementation of the bag of words extracts features within the images and condenses them down to a vector of 500 words, where each image is represented as a number of these words. The actual classification is completed by an SVM. The default parameters were used for this testing.

For the prototyping environment only a small data-set was gathered, as the goal was to test the validity of training a classifier on these images and determining if a correct prediction can be obtained, without worrying too much about the accuracy. Roughly 40 images were used to train the classifier.

4.3 Mobile Development

The next step of development was to utilise the learning and understanding that had taken place through the prototyping process and apply that to the mobile platform. This platform had some hardware differences that needed to be considered. Firstly, the size and potential brightness of the screens were lower then that of a laptop screen. Secondly, even though the processing power on phones has increased dramatically in the past few years, the typical phone is still behind that of a laptop in its processing speed.

The development process on the mobile platform was done in Android Studio. Even though this environment is not as conducive for rapid development, and there was less familiarity with the platform, the actual development time was considerably less than working in MATLAB.

4.3.1 Screen-Based Light Source

The development of the screen based light source, on the mobile device, took a very similar approach as previously taken. A very simple pattern was chosen, with half the screen being illuminated and half being dark (see Figure 4.12).

There are a few reasons for using this design. By having as much of the screen illuminated for each pattern as possible then the resulting signal is larger, hence the SNR is also larger. This can also be solved using multiplexing, where more light sources are added to each pattern. This can then be demultiplexed, by comparing the resulting lighting [42]. This approach would very likely produce better results, though the answer to which lighting patterns should be used to produce the good results is a non trivial one. Research has been done on this topic [43], but for this thesis it is not part of the question being investigated, so the simple pattern was chosen, which is known to work, whilst also having a large SNR. For the demonstration of this system there are only 3 different lighting patterns. Ambient light (none of the screen illuminated), lighting pattern 1 (left half of screen illuminated) and lighting pattern 2 (right half of the screen illuminated).

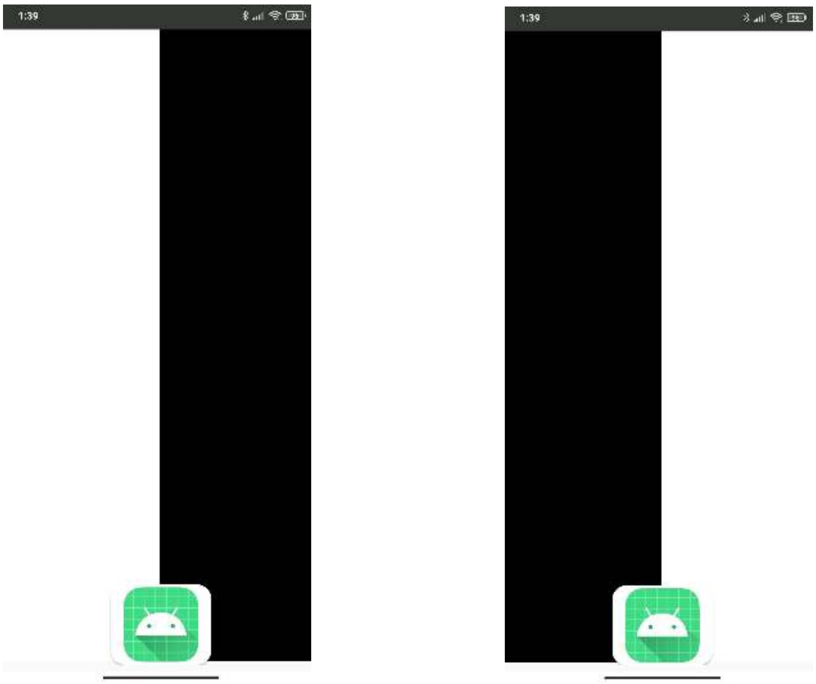


Figure 4.12: Image of lighting patterns which are displayed on the mobile screen.

4.3.2 Image Gathering

The image gathering process required the synchronisation of the image capturing and the lighting. The synchronisation is different for each platform and potentially each device. This is an important step to optimise as it means that the time required to complete the image capturing is reduced. A reduction in the imaging time means there is less time for the face being imaged to make any movements. It also reduces any changes in the background or ambient light that might effect the result because the imaging process takes in the vicinity of 30 images in under half a second. It is very difficult to analysis exactly what is occurring when each image is captured. To resolve this, a mirror was placed in front of the phone while imaging to reflect what the screen is showing at the time the image was taken. Then looking back through the set of images it can be seen how the time can be adjusted to ensure each image is being captured at the right time whilst reducing the overall time taken (See Figure 4.13).

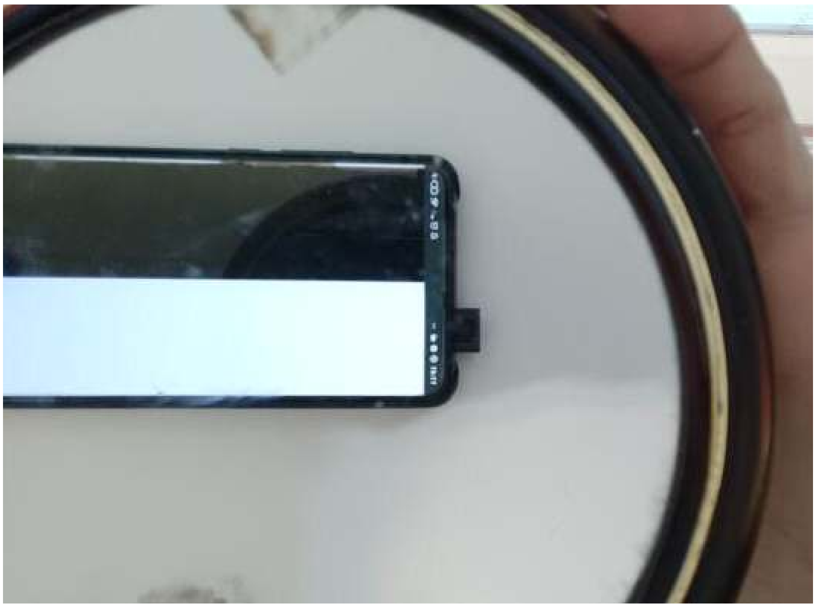


Figure 4.13: Image capturing with a mirror to reflect the screens state at the time of imaging

The physical imaging set up is much the same as that for MATLAB. One difference is because it is a smaller hand held device it can and will be held closer to the users face, so this was also done when imaging of a face occurred.

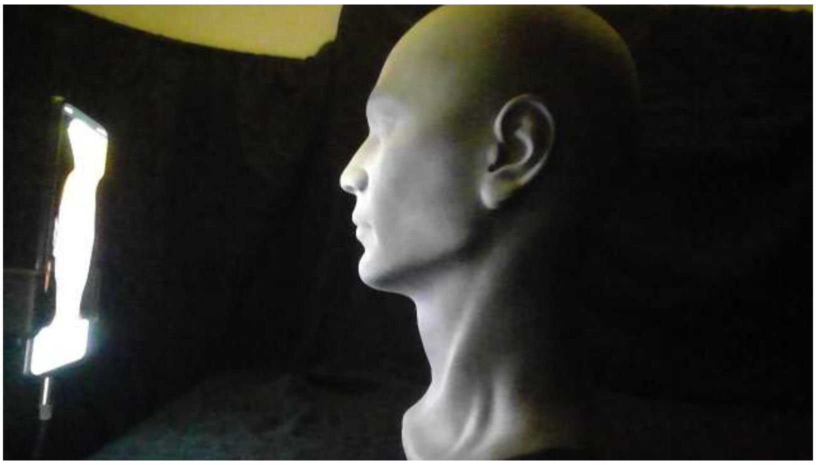


Figure 4.14: Image of physical imaging process.

4.3.3 Image Processing

The image processing that was completed on the mobile platform uses the exact same approach as developed previously. The image processing was completed using the Java implementation of OpenCV. The results achieved were slightly different because of the new platform. These are shown and explored below.

Image Registration

As discussed earlier, rigid transformations were more reliable than non-rigid as there was no way for it to degrade the signal with the transformation process. Rigid image transformation was applied to each of the images captured before any of the other image processing could occur. This was achieved using the preexisting image registration function in OpenCV. Below, in Figure 4.15, is an example of the image registration results.

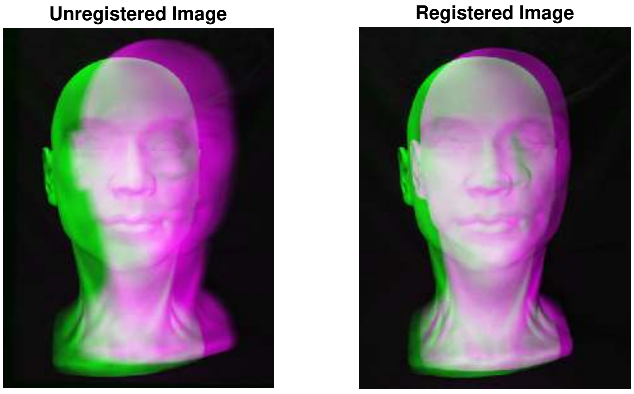


Figure 4.15: Rigid image registration from mobile images, using OpenCV function. Unregistered (Left), Registered (Right).

In this example, the movement was exaggerated. The image alignment algorithm corrected most but not all of the movement. In the atypical imaging process the overall image movement is typically less and the algorithm performs better.

Averaging & Noise Reduction

The image averaging on the mobile images is just the same process as done before, a simple mean of each pixel with the others in the image group; averaging all ambient light images together, then the first lighting pattern together and finally the second lighting pattern together. In Figure 4.16 it shows the effect of averaging. An investigation into how the different number of images being captured effects the results is shown in Section 5.

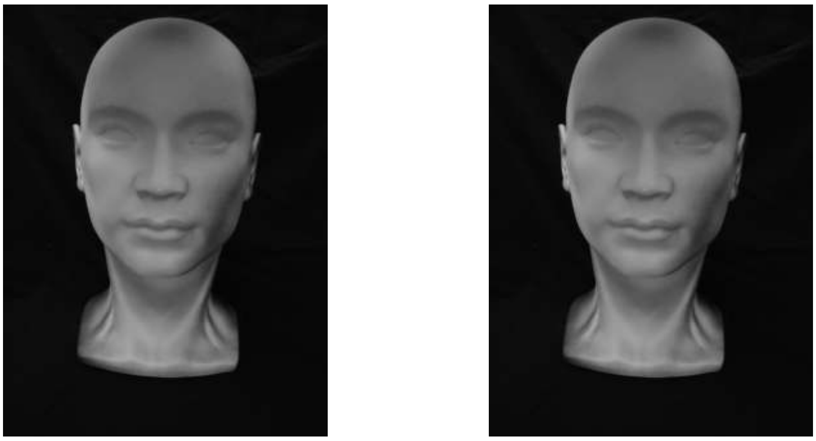


Figure 4.16: Example of noise reduction using averaging on mobile images. The images are: One image captured (Left), Ten images captured (Right).

In these images there is no visual difference as a result of the averaging, although the effect of this process becomes prevalent by the end of the process. See Section 5.3.2 for more details of the overall effect.

Ambient Light Removal

The ambient light was removed by sidetracking the averaged light pattern images from the averaged ambient light images. Because of the different camera used and the difference in light produced from the screen, the resulting images are slightly different to that shown from the laptop environment.

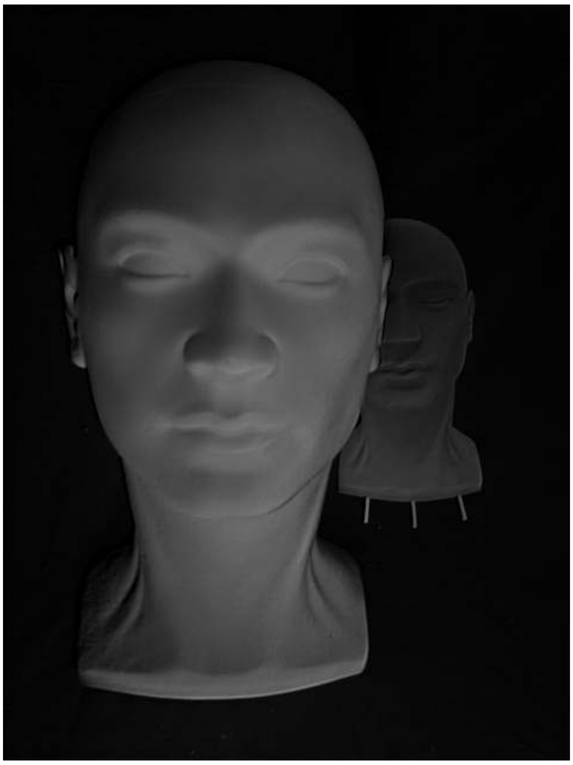


Figure 4.17: Results of ambient light removed from averaged mobile images.

Image Manipulation

The process to remove the albedo term from the images is the same process as was performed in the laptop environment; by dividing separate lighting patterns. The result is shown in Figure 4.18

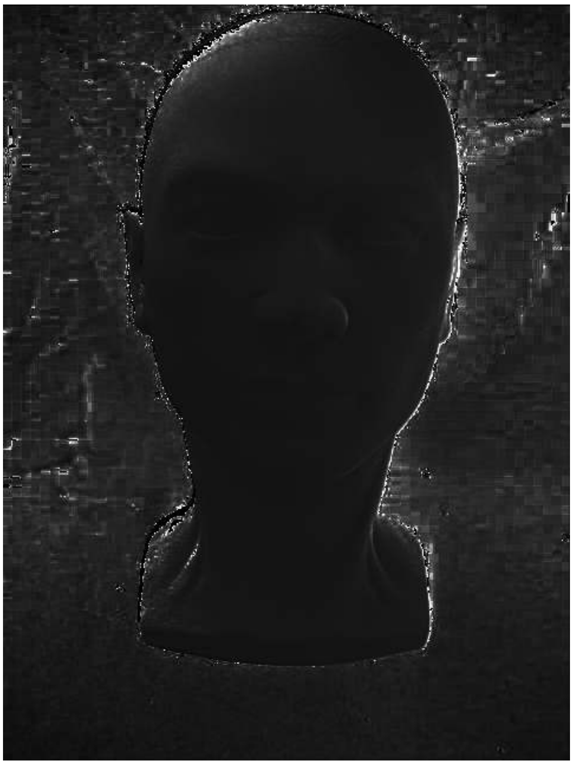


Figure 4.18: Image with albedo values removed, indicating surface normal values. This image was taken in low light conditions.

The noise around the image can easily be seen. This is a result of the division process when small values are multiplied together. Filtering is applied to be address.

Filtering

The filtering that was performed used the same process as described in Section 4.2.3. Even though it performed worse than what was done in MATLAB, depending on the ambient lighting conditions, this could be in part due to the different camera being used but also in the reduction of different lighting patterns from 4 to 2 meant that that filtering had less information to go off. It was more difficult thresholding what is the foreground(face) and what is the background. The example shown here was completed in low light conditions which allowed it to perform well.

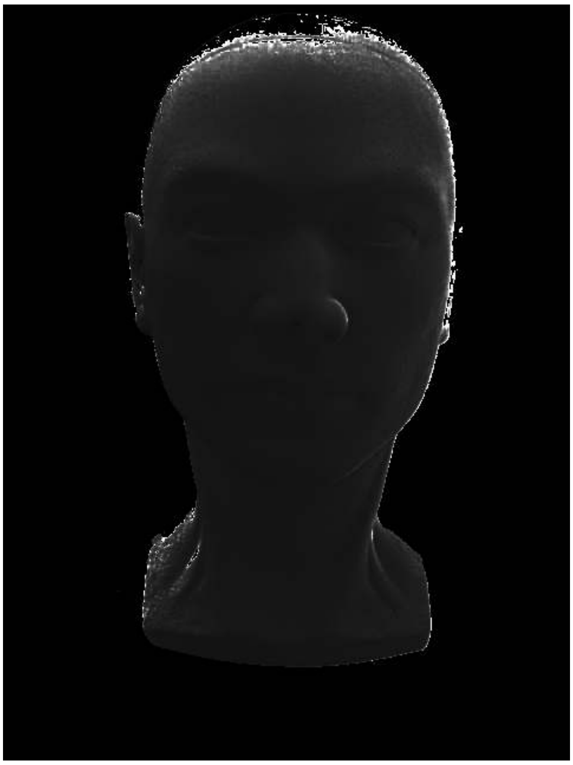


Figure 4.19: Filtered mobile image, to remove the background

4.3.4 Classification

The implementation of the classifier on the mobile device was done a bit differently to that in MATLAB. The way in which classifiers are implemented on a mobile application is to pre-train a model, then import the model into Android Studio where it will get used to predict the images classification. Because Android is owned and run by Google, there is a lot of support within Android Studio for working with TensorFlow classifiers because they are also owned by Google. Specifically, TensorFlow Lite models as they have been designed to run on lower power devices like phones.

For simplicity, TensorFlow Lite Model Maker was used, which allowed for the image classifier to be easily trained on a data set. Because of its simplicity it does remove some of the customisation and tailoring to the specific application; but for this system, having a simple solution that works is satisfactory. The classifier was trained on around 500 images under a variety of different ambient lighting conditions.

4.3.5 User Interface

A basic user interface was developed for the application to allow for testing to take place. This involved showing a preview of the camera view, to show exactly what was being imaged. An initiate button was also made to start the image capturing and processing.

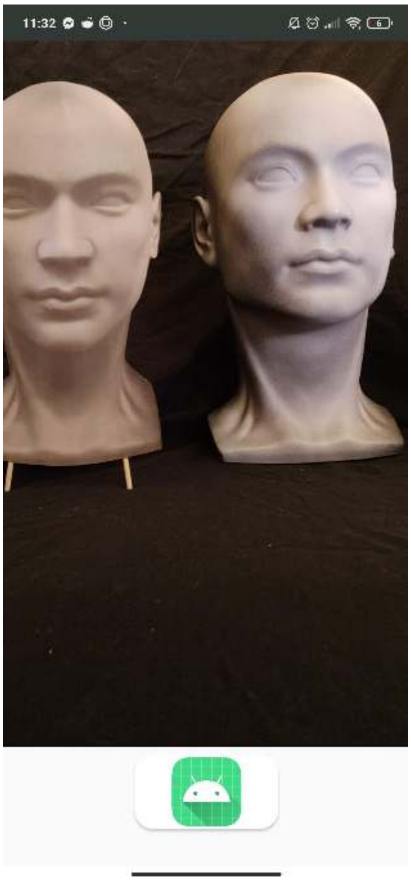


Figure 4.20: The user interface made to allow capturing and processing.

To show the user the results from the image processing and classification, a simple icon was shown on the screen. A green tick was shown when a real face(in this case the model mannequin head) was predicted, where as a red cross was shown when a fake face, in this case a printed picture of the model mannequin head, was predicted. Figure 4.21 shows both cases.

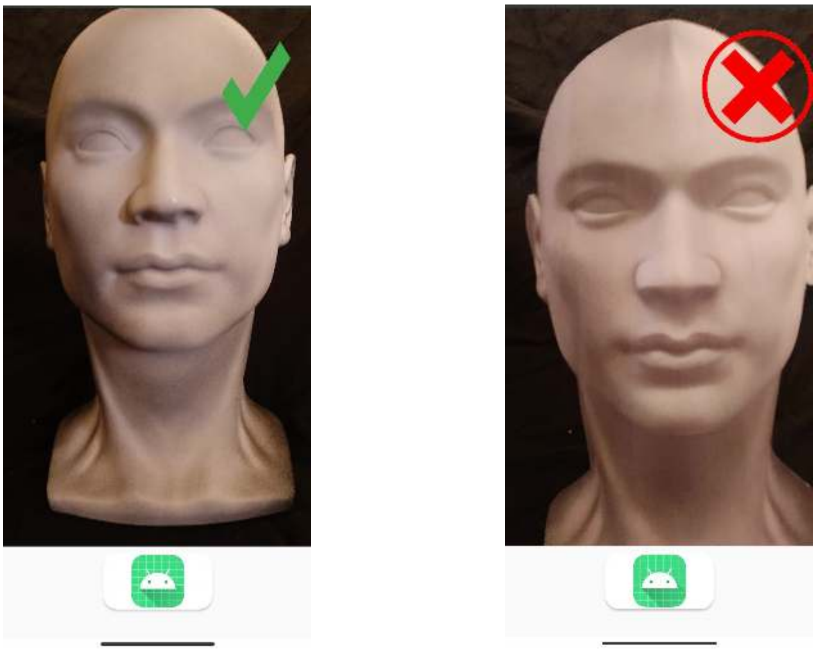


Figure 4.21: Accept or reject symbol is displayed on the screen to show the classifiers results.

4.4 System Evaluation

The systems benchmarks and evaluation were performed in MATLAB, with images gathered from the mobile platform. Even though the mobile system has its own classifier, performing the classification and evaluations on a saved data set of images in MATLAB will be easier to manipulate and record.

For this evaluation 500 images were captured and saved from the mobile system. These images were then exported into MATLAB. They were then used to train the classifier. The classifier is the same as that used in the laptop environment, it is using a Bag of Words structure, with MATLAB's pre-built functions. The performance of the system under differing conditions, such as ambient light or movement, is evaluated against this classifier.

Results 5

In this chapter the results from the method followed in the previous chapter are shown.

5.1 Prototyping Results

5.1.1 Imaging Processing Results

The results gathered from the computer platform have been laid out in the general progression in which the results were gathered and then built upon. For each image that was captured using the light stage, the light pattern used is a simple lighting of the left half, right half, top half, bottom half of the screen.

5.1.2 Noise Reduction by Averaging

This noise was reduced by taking multiple images and averaging the value recorded at each pixel across all the images. This can be done because the noise acting on these images is assumed to be unbiased and therefore, it will converge to zero with enough averaging of images. This significantly increases the number of images and time required to perform the image gathering, which also comes with diminishing returns. Finding the best balance between required time and noise level requires taking many images and comparing their values.

If we assume that averaging 200 images is fairly close to true values, at least in terms of what can be removed by averaging, then an estimate of the noise present with each number of averaging can be guessed. Plotting the Root Mean Square

(RMS) shows how it changes as the number of samples change.

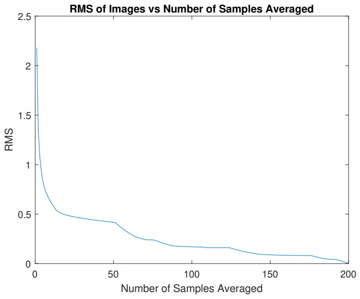


Figure 5.1: Plot of RMS values over number of samples

From Figure 5.1 it can be seen that about 80% of the reduction in RMS is done in the first 15 sample averages. The process used to generate this plot was not completely precise but it allows for a rough indication of how many samples to take.

Ambient Light Removal

The first metric that can be shown is the efficiency of the ambient light removal from the images. This allows the signal to be extracted from the image. With larger amounts of ambient light, the signal within the image is relatively smaller, meaning the SNR for this case is smaller. Below is an image of ambient light being removed at different ambient light levels.

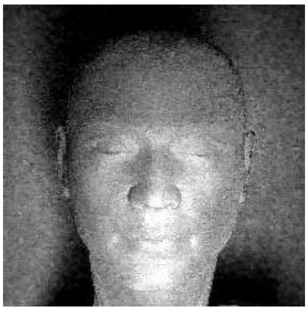


Figure 5.2: Low ambient light removed

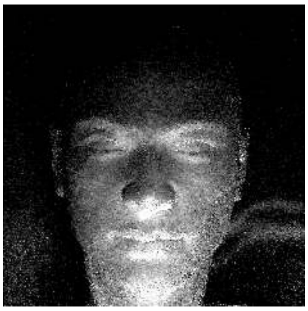


Figure 5.3: High ambient light removed

The images are processed to expand to the whole brightness spectrum, to represent the difference more. This is demonstrated in both Figure 5.2 and Figure 5.3. Significant noise can be seen within each image because amplifying the small signal of light within each image also expands the noise within it. The noise in these images needs to be reduced.

Removal of Surface Albedo

Referring back to Equation 4.3, the relativity term is mathematically removed by the division of two different lighting patterns, this has been implemented and was run comparing the patterns, left half with right half and top half with bottom half. This process allows the surface normals to be extracted, meaning that the model head can now be compared to a printed image version.

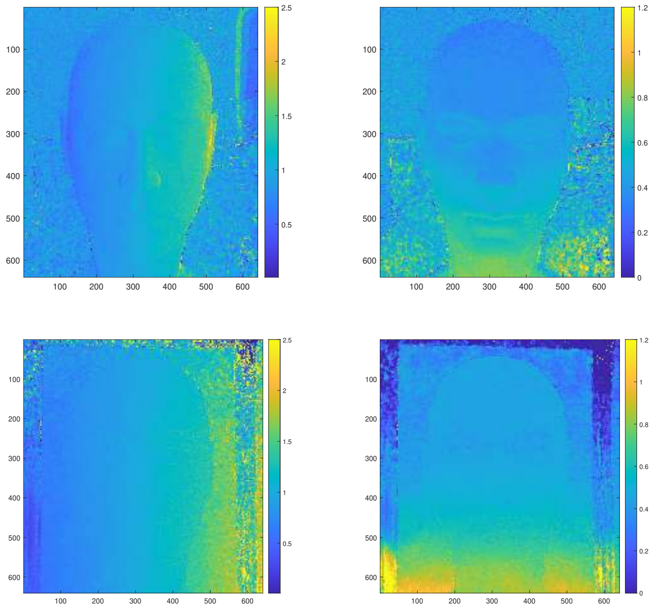


Figure 5.4: Reflectively term removed, Model head (top) vs Printed head (bottom)

This image has been plotted with a colour spectrum so as to visually see the difference across the image. Because of the nature of dividing two numbers, some pixels would increase dramatically, or be divided by zero and produce either *NaN* or *Inf* values within MATLAB. This image has been limited so that values that would be above 2 are capped at 2.

The difference can be clearly seen between the model head and the printed head under the exact same conditions. With these differences a classifier could be trained to recognise the difference between these two test cases.

Image Filtering

Because of the amplification of small values when the division occurs, noise within these pixels is amplified, this means filtering is necessary. Applying the logic explained in Section 4.2.3 the following image was gathered.

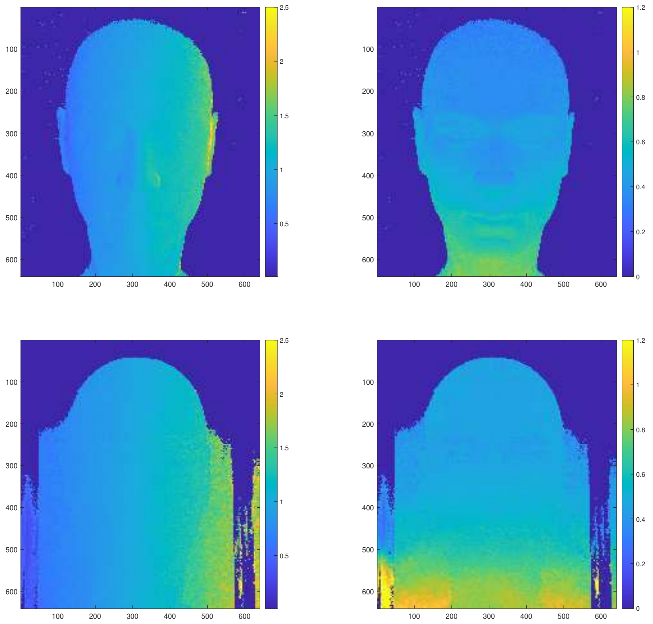


Figure 5.5: Reflectively term removed and filtered, Model head (top) vs Printed head (bottom)

The threshold value for the filtering was adjusted and the result from this gave an image with only the area of interest still remaining. Unfortunately, the threshold value, which produces the best filtering, depends on the conditions the image was taken in. This means the current filter is not a universal solution. This will need to be fixed moving forward.

5.1.3 Classifier Performance

The classifier was trained using images of the model head and images of the printed head. The images were taken under different lighting conditions and positions relative to the camera and light stage. This meant that the filtered images were not used because it did not give good results across all the images. Figure 5.6 and Figure 5.7 show examples of the images within the data-set.

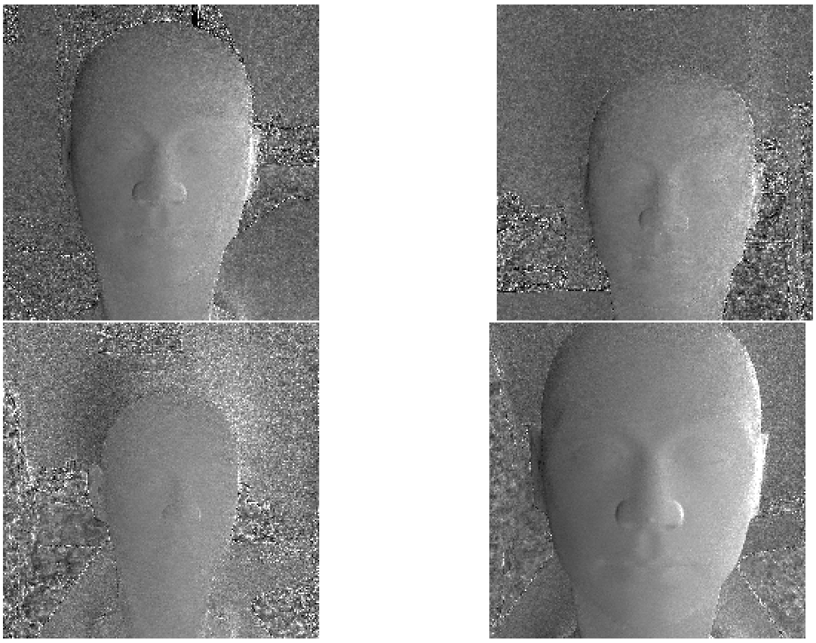


Figure 5.6: Model head images used in testing

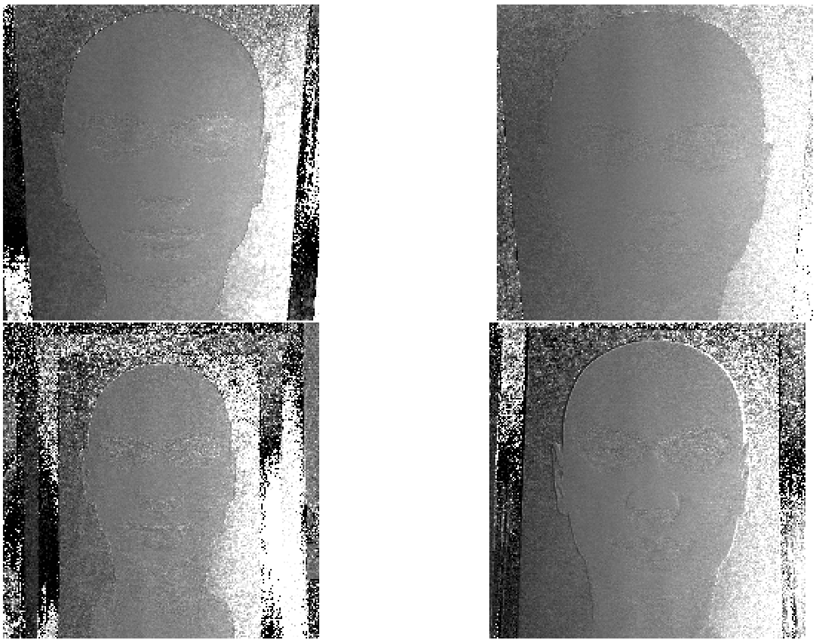


Figure 5.7: Printed head images used in testing

Another eight images were gathered in varied lighting and positions, half of the model head and half the printed head. The classifier was run against each of these images. It correctly identified each image, with 100% accuracy. Even though this was a fairly small sample with limited scope, it confirmed that the methodology is valid.

5.2 Mobile Images

5.2.1 Imaging Processing Results

The final image produced on the mobile platform, was run on the real model head and the printed version of the model head. There was large differences in the final images for each case. Below in Figure 5.8, images were taken in low light conditions for each case.

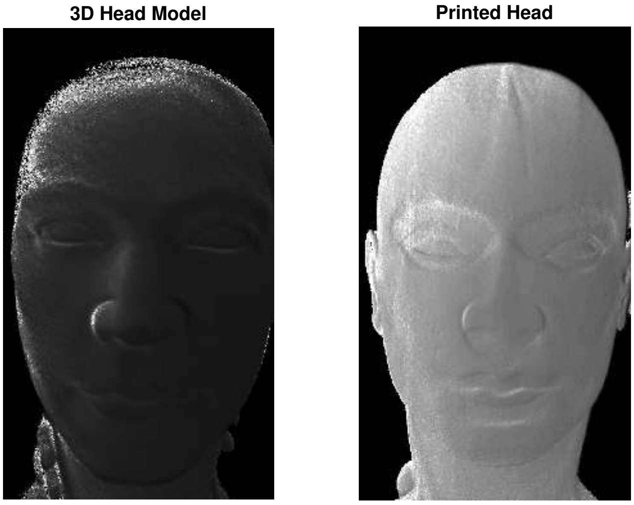


Figure 5.8: A comparison of the real and fake version of the model head running on the mobile platform.

Below are additional images for both real and fake faces. These are the raw images which have been outputted by the mobile application, without any contrast adjustment or realisation.

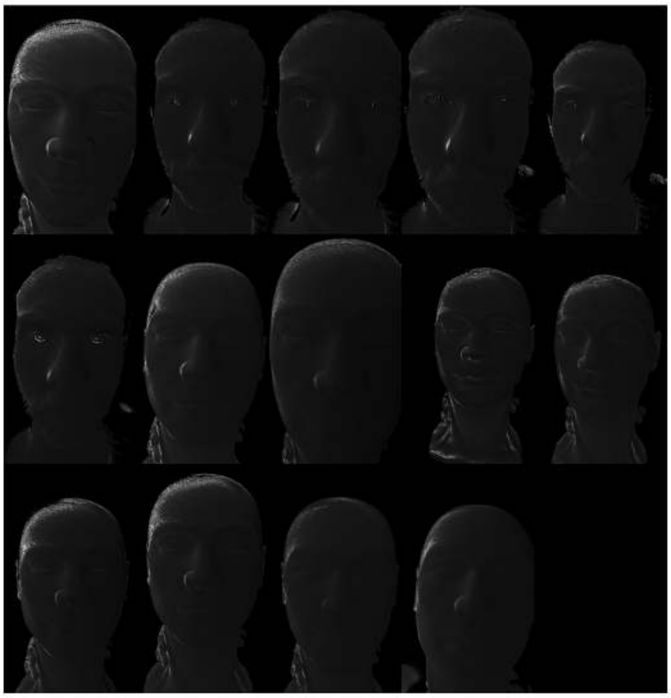


Figure 5.9: Montage of real face images processed by the mobile application.

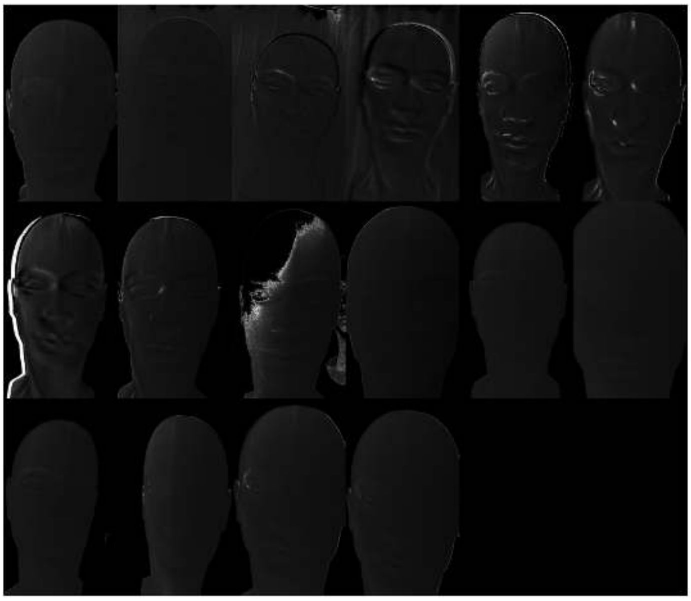


Figure 5.10: Montage of fake face images processed by the mobile application.

It is hard to define, however, in the 2D face images almost all the features within the face have been removed and it just looks like a flat object. Meanwhile, the 3D model head still shows some information about the structure of the head.

5.2.2 Classifier Performance

The data-set gathered of around 500 images was split into a training and evaluation set. The split was 80% to 20% for training and evaluating respectively. Using the evaluation set the performance of the classifier was found. See Table 5.1.

Table 5.1: Confusion matrix of classifiers performance

		Prediction Face	
		Fake	**Real**
Actual Face	**Fake**	82%	18%
	Real	0%	100%

The overall accuracy of the system is:

$$Accuracy = 91\%$$

All of the real faces were correctly classified whilst only 82% of the fake faces were correctly classified meaning 18% of the fake faces were seen as real faces.

5.3 Test Scenarios

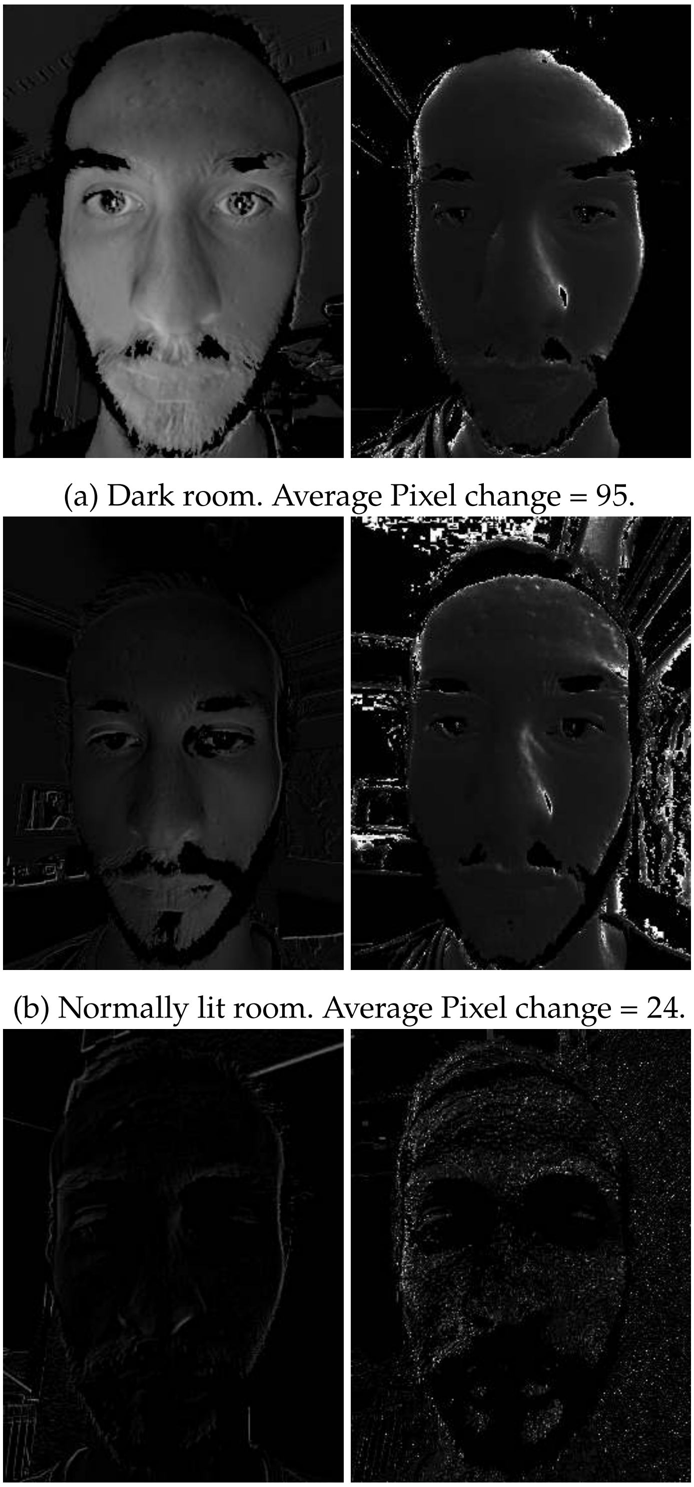
As there are many different factors which effect the final performance of the system it needs to be investigated exactly what changes they make. These evaluations have been done using the classifier described above.

5.3.1 Ambient Light Level

Roughly 500 images were captured in a range of lighting conditions. For the gathered images, 5 images were gathered for each lighting pattern. This accumulates to 15 images in total. The method, in which ambient light level is measured numerically, is to calculate the average pixel value in the ambient light removed image, as this value is directly proportional to the amount of ambient light.

Produced Images

Here the intermediate (ambient light removed) and final (reflectively removed and filtered) images are shown for 3 different light levels.



(a) Dark room. Average Pixel change = 95.

(b) Normally lit room. Average Pixel change = 24.

(c) Sun light. Average Pixel change = 0.65.

Figure 5.11: Resulting images at different lighting levels.

From the images it is obvious that as the ambient light increases the results deteriorate. Starting with strong results in low light conditions, then ending with very poor results in direct sunlight.

Performance

The data-set gathered at each differing ambient level, consisting of 500 images in total, was evaluated against the pre-trained classifier. The overall accuracy of the data-set has been plotted against the average ambient light level for the data set, using the method described above to obtain the ambient light level.

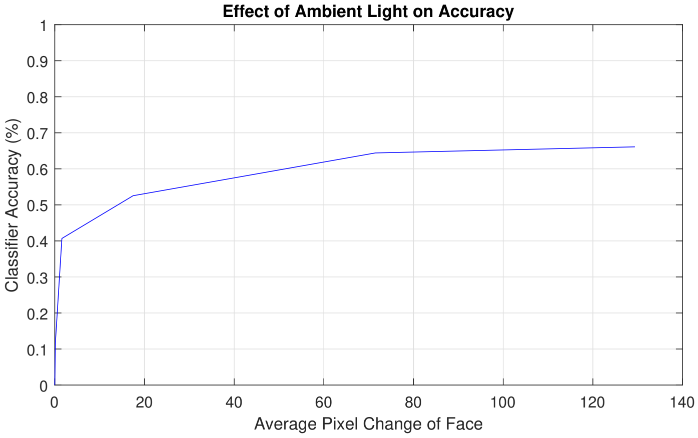


Figure 5.12: Graph of the overall performance against the ambient light level.

This graph conforms numerically what was seen in the image, the systems ability to classify real and fake faces has a strong correlation with the ambient light at the time of imaging.

As a reference, when viewing Figure 5.12, the following table shows what the average pixel typically correlates to in the real world.

Lighting Condition	Average Pixel Change
Lowly lit room	95
Normally lit room	24
Sun light	0.65

Table 5.2: Real world representation for different values of average pixel change.

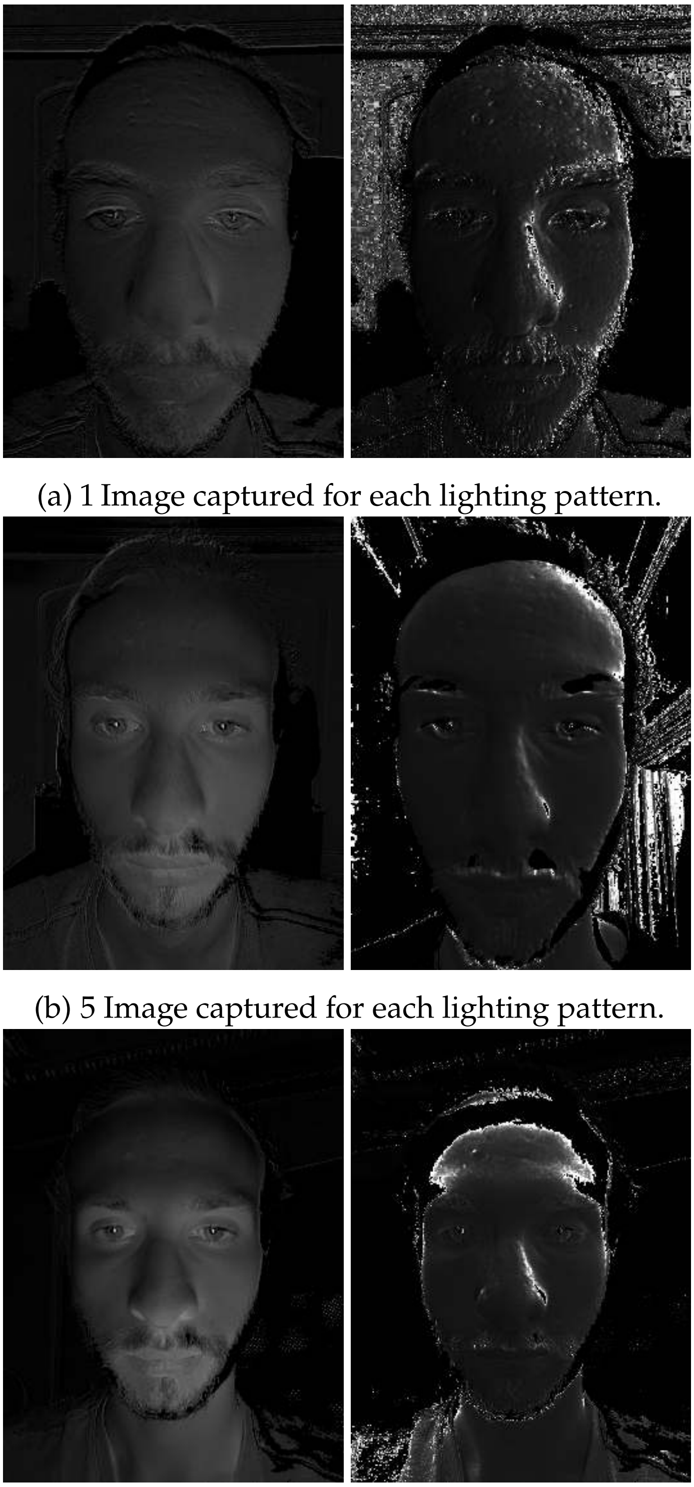
The reason for the lower accuracy compared to that seen in the previous section, is likely two fold, as a large data-set was required, the number of averaged images for each case was less. Also, there were more fake faces in the data-set then real faces, as can be seen in Table 5.1, they are not classified as well as real faces. This graphs shows the relative performance at different light levels.

5.3.2 Number of Images Captured

The number of images captured can dramatically reduce the noise that is present in the final image. This section investigates this relation to its overall classifier performance. Images were captured with a different number of images per light pattern. Following the same approach taken in the previous section, 100 images were taken with 50 being a real face and 50 being a fake face. The number of images captured per pattern are as follows; 1, 2, 5, 10 and 20. The total number of images captured for each reading will be 3 times that amount as there are 3 patterns used.

Produced Images

The images shown are the ambient light removed images and the final filtered image.



(a) 1 Image captured for each lighting pattern.

(b) 5 Image captured for each lighting pattern.

(c) 20 Image captured for each lighting pattern.

Figure 5.13: Resulting images as the number of images captured increases. The left column is the image with ambient light removed. The right column is the final images with albedo removed.

The the difference in the ambient light removed images is quite small, though when looking at the final image there are noticeable differences. When only one image is averaged the final image exhibits a lot of noise. Interestingly, when twenty images are averaged together, the filtering starts to encroach on the face.

Performance

The overall performance of the system, for differing numbers of images captured is plotted below.

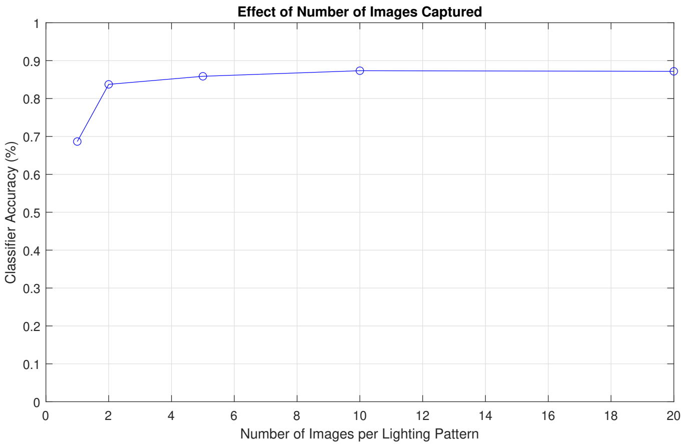


Figure 5.14: Graph of the overall performance against the number of images captured.

Initially increasing the number of images drastically increased the performances but as more images were added the benefits to the classifiers performance tapper off. In fact, when 20 images are captured it actually has a worse performance.

The worse performance and worse filtering when 20 images are captured is likely due to the fact that that the exposure of the camera is not fixed, which means the longer capturing time gives the camera time to adjust to the new brightness. Which in turn means that one light pattern is lighter than the other.

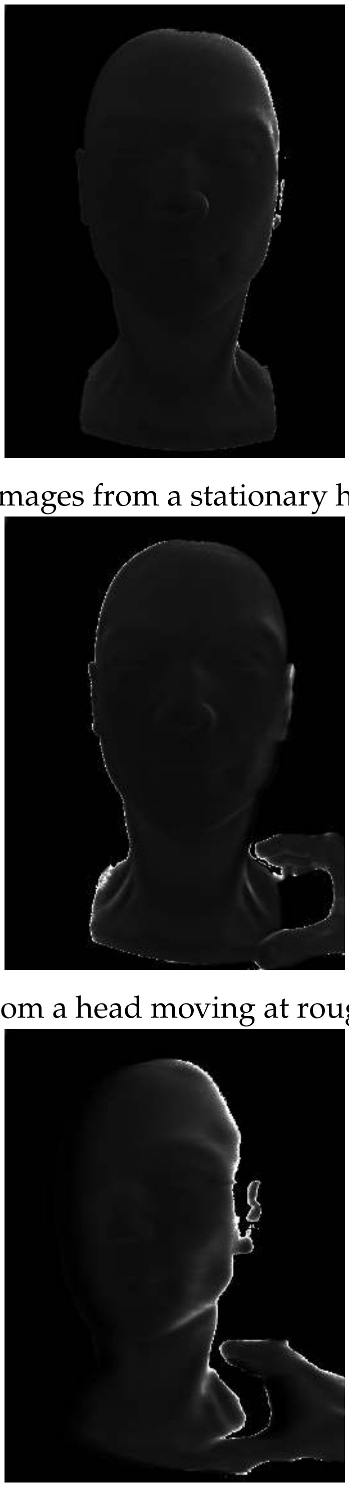
5.3.3 Movement Between Images

The amount of movement within the image effects the ability of the system to correctly classify faces. This has been mitigated to some extent by the image registration process that was implemented, though given enough movement there will be some misalignment between the images. The results below investigate how different levels of movement effect the image process. The different levels of movement have been quantified using a rough estimate of velocity. This velocity number is not indicative of all the nuances that can occur within the movement but it will give good relative results between the levels. 100 images were captured, half real and half fake faces, at 4 different movement levels, the movement was varied across the image sets, such that it should encompass the different types of movements a user can make while imaging.

In this case, the model head was used when imaging instead of using a live head, as it allowed for move control over specific movement levels. This would not be possible with a live head. A tripod held the phone so that only the head movement could be isolated.

Produced Images

The images shown are after the image alignment and average process is performed.



(a) Images from a stationary head.

(b) Imaging from a head moving at roughly 0.02 m/s.

(c) Imaging from a head moving at roughly 0.10 m/s.

Figure 5.15: Resulting images at different movement levels.

As expected, with increases in movement the resulting images no longer representing the surface normals of the face. With increasing movement the features on the face are blurred together. At 0.02 m/s the movement is corrected but blurring still takes place. At 0.10m/s the result is almost at the point were it is no longer recognised to have the structure of a face.

Performance

The 100 images captured at each level were evaluated individually, the overall performance of each data-set has been plotted against the movement speed.

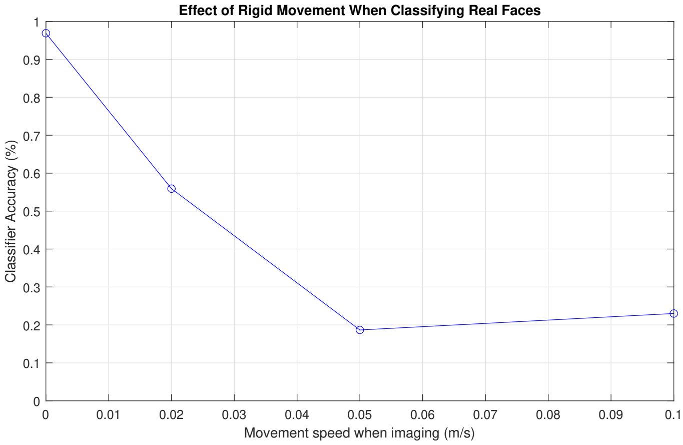


Figure 5.16: Graph of the overall performance against the amount of movement between images.

There is a strong correlation between the classifiers performance and movement. This shows that even with the image alignment being done the movement effects the results. In this case, the alignment algorithm which has been utilised in the mobile application hasn't performed very well. It goes to show that having a strong image registration process is crucial for this system.

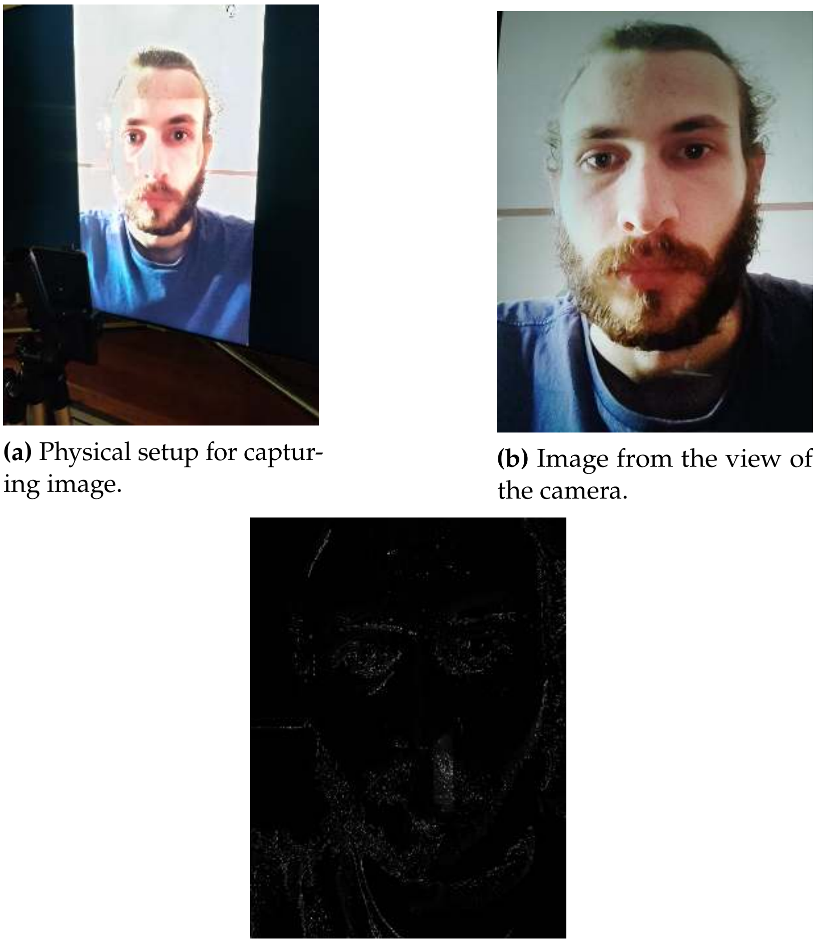
Based on the images taken, an estimation for the typical amount of movement that is present when imaging a live face would be around $0.01m/s$, for a person who is trying to stay still. Using Figure 5.16, this amount of movement negatively impacts the overall accuracy but it is at an acceptable level before the results become just an uneducated guess, which has occurred when excessive movement is present.

5.3.4 Overall Time vs Accuracy

As the time increases to perform the image processing, whether that be through additional images captured, longer image registration, etc., the accuracy improved. This improvement observes diminishing returns where at a certain point the increase in time no longer produces a worthwhile increase in performance.

5.4 Proposed Solution Working Over Existing Solutions

Existing solutions can be fooled quite easily with an image of the user's face. Situations were found in which existing solutions deployed on phones can be fooled, where as the method developed in this thesis was not. Figure 5.17 shows an example situation in which the proposed solution works better than the existing one. The situation shown is an image of a face being displayed on a large TV.



(a) Physical setup for capturing image.

(b) Image from the view of the camera.

(c) Output after the imaging processing.

Figure 5.17: Existing solution being fooled, while the proposed solution is working correctly.

In this example it has been shown that, existing methods can be improved by the proposed method.

The results produced showed that the system behaves differently when imaging a screen. Because the screen itself produces light and also has a refresh rate where its own light is fluctuating, it is likely effecting the results. This is not necessarily a bad thing as it means the results look nothing like a real face. From this it can be speculated that screens will have a hard time fooling the method developed.

Discussion 6

This chapter discusses the implications of the results achieved and the learning from the development process.

6.1 Accuracy Considerations

The system developed has a strong dependency on the imaging environment and the subject. For a security system this is not ideal, but with further development this approach could be adapted to compensate for these dependencies.

6.1.1 Ambient Light & Number of Images

As shown in Chapter 5, there is an inverse relationship with the ambient light level and the performance (decreased SNR) achieved. It was also seen that there was a relationship between the number of images taken and the performance (increased SNR).

These relationships expressed together show that there is an optimal number of images which can be taken to obtain the required SNR for peak performance. This optimal number depends on a few factors but chiefly it depends on the ambient light. In high ambient light conditions a large number of images are required to increase the SNR. While in low light conditions a much smaller number of images are required. Increasing the SNR above a certain level provides diminishing returns of performance increase. Taking the same number of images in low light as is required for high light conditions adds a lot of time with little benefit. Ideally, an adaptive approach would be taken, where the system will determine the required images based on the observed ambient light.

6.1.2 Movement

The movement within the images is not an SNR problem that can be solved by taking more images to reduce noise. This issue instead deteriorates the signal, so that it no longer represents the true make up of the face with the surface normal values. Instead because of the misalignment of the images, the desired signal can no longer be extracted from the ambient light, meaning the image not longer represents the surface normal values of the face.

In the testing completed, excessive movement was the cause of many incorrect classifications. There are ways to mitigate the effect that movement has on the performance of the system, on top of having a good image registration algorithm. One option is to perform an additional check between the captured image to determine the change within each frame. If the movement is beyond a certain threshold then the image would not be considered for the classifier.

6.2 Screen Based Light Sources

As discussed in Chapter 3. Having a screen based light source is a fairly novel idea. This thesis has investigated one implementation and application of this idea.

6.2.1 Comparison to Existing Light Stage & Photometric Stereo Solutions

Comparing the developed system to that of existing solutions, which were discussed in the literature review, shows some difference in the results. These existing, highly sophisticated solutions can produce results which are a lot better then what was observed with the developed system. This is to be expected though as they have not been designed to be portable and easily accessible, instead they are focused on increasing performance. The approach compared to the other screen based lighting solution has been tested for very different use cased. When the developed system was used on static objects and in low light conditions, the results obtained were somewhat comparable to the existing solutions. Though in there case they were interested in getting the exact surface normal value instead finding values relative to the surface normals.

6.2.2 Operating in a Non-Controlled Environment

The system performed quite well considering that it was designed with the goal to work in a wide range of environments, instead of just a controlled lab setting. The developed solution, has been shown to be extremely versatile for a photometric stereo method. From the literature researched, the existing solutions were not usually developed for the range of environments that this solution was. As a result the system was able to cope with a ambient light and movement being present in the imaging process. Though a large amount of either one would drastic reduce the maximum performance that could be achieved.

6.3 Mobile Facial Detection

The performance of the system to classify real from fake faces was fairly reliable. Although for a security system the results and testing have be extremely tenuous, as reliability of the system has to be proven before it can be deployed. Currently, the proposed method has shown a lot of promise, but it has not been put through the required testing to make sure it is a reliable solution.

6.3.1 Comparisons to Current Solutions

The developed solution has been proven to work in situations in which an existing solution, deployed on phones, does not. This means that the hypothesis outline in the introduction holds true. Facial recognition systems can be improved, with the proposed method. In comparison to the facial recognition techniques which use multiple sensors such as that on the iPhone,this method needs further development to improve its performance.

The time required to perform the image capturing and classify of real and fake faces for the current system is about half a second. This is within an acceptable amount. However, compared to the existing solutions, the time they require to perform there facial detection is undecidable from the time the phone is picked up.

6.3.2 Additional Layer of Security

Because the proposed solution is to work as an additional layer of security on top of existing solutions, certain considerations can be made from it. The first point to make is that adding another layer of security, more information, will never make the system perform worse then without it. For the solution to have a use case it doesn't have to be perfect. Instead, the trade off that the solution needs to make is whether the performance boost is worth the extra computation/capturing time as well as any impracticalities that arise from it.

The amount of time that is required to perform the facial recognition will be the addition of both the methods. The reason this is an important consideration is that if the time required for the user to unlock there phone is longer than putting in their passcode or scanning their finger then the use cases for this application are drastically reduced.

6.4 Limitations

There are some fundamental limitations to the approach outlined in this thesis. Because the approach has the desired signal embedded in with the ambient light; if the amount of light that the screen adds to the scene is very small, compared to that of the ambient light, it means that when the ambient light is removed the resulting image will be very noisy. This is because the gain will have to be increased dramatically to account for the small signal. This error can be mitigated by the noise reduction process of averaging multiple images together. The more ambient light, the more images that need to be added together. Even though theoretically this relationship would hold for any level of ambient light, practically it does have a limit where the number of images required to extract the signal from the noise becomes excessive. For this application this occurs in direct sunlight. The amount of light the screen adds is just too small to pick up on.

This relationship between the ambient light and number of images required also extends to the amount of time that is required to capture and process the images. If there is a lot of ambient light and 200 images are required to reduce the noise enough then the time required to perform the classification is drastically increased to a case where only 20 images are needed. Regardless of the number of images, this process is always going to take longer than the existing approaches as they often take a single image. Though the extra amount of time is definitely not prohibitive for the method.

The method also relies heavily on the existing hardware and the performance could vary drastically between different devices. The approach was taken in such a way that it should be able to work on almost all devices with a screen and a front facing camera. However, the actual quality of the camera as well as the brightness and darkness of the screen will effect performance. The testing of this system was done on an OLED screen which doesn’t have a full back light. This means the dark sections of the screen can be darker than what is usually possible with an LCD screen. Meaning that OLEDs will be able to produce greater contrast in signal generation and, thus, will likely perform better.

Conclusions & Future Work 7

7.1 Conclusions

This thesis investigated the ability to have accurate facial detection on mobile phones that only have a single camera sensor. The method in which it was proposed to achieve this was to use the techniques developed in photometric stereo and light stages, while using the phone screen as the light source.

7.1.1 Single Camera Facial Detection

It was shown that a single camera facial detection system can be improved, without the need for additional sensors. This was done by investigating the use of the screen based light source. As the system that was produced is to work in conjunction with other approaches it would need to be combined with these existing solutions before it could be deployed because together they would be able to complement each other's weaknesses to provide better accuracy. If this was to be done, then it has the potential to increase security on 80%-90% of the phones which provide facial detection as a security method.

7.1.2 Screen-based Lighting Applications

What was made clear with this project is that using a screen as a light source for photometric stereo or a light stage is a viable method. As stated earlier, this screen based approach has not been explored thoroughly within relevant literature, and there are many applications in which this approach can be used to make a portable, accessible version. There are many applications within the entertainment industry where special effects need detailed images of actor's faces for post lighting/production. Other applications could include, a readily available 3D scanner to model objects, or classification of objects for the visually impaired. There is a lot of potential in this area because of how available screened devices are within society.

7.1.3 Thesis Contributions

The main contributions made by this thesis are stated below.

- This thesis has shown that is it possible to improve the workings of existing single camera facial recognition systems, with the approach outlined in this thesis. There are situations in which the conventional method developed on phones fails, while the method in this thesis doesn't.
- Using a screen based light source for different photometric stereo and light stage applications is a viable option. As shown by the results of this thesis.

7.2 Future Work

There is a lot of potential future work that can be done on this topic. Firstly, there should be a more extensive evaluation process to evaluate the system. The testing performed in this thesis shows a working system in fairly isolated conditions. There are questions of how well this method will perform on other hardware, not just the single sensor phone it was tested on. One of the main goals of the proposed approach was that it should work on a range of devices, to see how different hardware configurations effects performance. Testing on a wider range of real and fake faces would also give a better indication of how well it would work with the general public and the effects that different facial features have on performance.

For the system to be practical and wanted to be used by the public as a way to unlock a phone the amount of time it takes to validate the face needs to be reduced. The theoretical limit for the amount of time needed is extremely low, with 60Hz screens and high frame capturing cameras. The process could be performed so fast the eye doesn't have time to register that the screen has been flashing. This would give the added benefit of reducing the misalignment between images as there is little time for movement to happen.

Finally, the combination of this method into an existing method would require significant investigation as to what system it should be combined with; such that their weaknesses complement each other so that the maximum accuracy is achieved. The existing solutions which were explored in the literature review, are usually movement based, such as blink detection, which is an ideal pairing for the proposed method. This is because these movement methods can be fooled by a video being played of the device owner's face, but not by a 3D mask of the face. Where as the proposed method in this thesis will not be fooled by a video as it is still a flat object. However, it could be fooled by an accurate 3D mask of the face.

Bibliography

[1] Samsung for Business. *Which Biometric Authentication Method Is the Most Secure?* Apr. 2020. URL: https://insights.samsung.com/2020/02/12/which-biometric-authentication-method-is-the-most-secure-2/ (cited on page 1).

[2] James T Kajiya. 'The rendering equation'. In: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 1986, pp. 143–150 (cited on page 5).

[3] Sukhjinder Kaur. 'Noise types and various removal techniques'. In: *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)* 4.2 (2015), pp. 226–230 (cited on page 6).

[4] Uğur Erkan and Adem Kilicman. 'Two new methods for removing salt-and-pepper noise from digital images'. In: *scienceasia* 42.1 (2016), p. 28 (cited on page 7).

[5] Robert J Woodham. 'Photometric method for determining surface orientation from multiple images'. In: *Optical engineering* 19.1 (1980), p. 191139 (cited on page 8).

[6] Paul Debevec et al. 'Acquiring the reflectance field of a human face'. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000, pp. 145–156 (cited on pages 9, 15).

[7] Derek LG Hill et al. 'Medical image registration'. In: *Physics in medicine & biology* 46.3 (2001), R1 (cited on page 9).

[8] Lisa Gottesfeld Brown. 'A survey of image registration techniques'. In: *ACM computing surveys (CSUR)* 24.4 (1992), pp. 325–376 (cited on page 10).

[9] J-P Thirion. 'Non-rigid matching using demons'. In: *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 1996, pp. 245–251 (cited on page 10).

[10] Theodore A Camus and Thomas A Chmielewski Jr. *Image subtraction to remove ambient illumination*. US Patent 6,021,210. Feb. 2000 (cited on page 10).

[11] Johan AK Suykens and Joos Vandewalle. 'Least squares support vector machine classifiers'. In: *Neural processing letters* 9.3 (1999), pp. 293–300 (cited on page 12).

[12] Harris Drucker et al. 'Support vector regression machines'. In: *Advances in neural information processing systems*. 1997, pp. 155–161.

[13] Eric Nowak, Frédéric Jurie, and Bill Triggs. 'Sampling strategies for bag-of-features image classification'. In: *European conference on computer vision*. Springer. 2006, pp. 490–503 (cited on page 12).

[14] Glenn M Fung and Olvi L Mangasarian. 'A feature selection Newton method for support vector machine classification'. In: *Computational optimization and applications* 28.2 (2004), pp. 185–202.

[15] Amna Sarwar et al. 'A novel method for content-based image retrieval to improve the effectiveness of the bag-of-words model using a support vector machine'. In: *Journal of Information Science* 45.1 (2019), pp. 117–135.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 'Imagenet classification with deep convolutional neural networks'. In: *Communications of the ACM* 60.6 (2017), pp. 84–90 (cited on page 12).

[17] Stephen O'Hara and Bruce A Draper. 'Introduction to the bag of features paradigm for image classification and retrieval'. In: *arXiv preprint arXiv:1101.3354* (2011).

[18] G Hemalatha and CP Sumathi. 'A study of techniques for facial detection and expression classification'. In: *International Journal of Computer Science and Engineering Survey* 5.2 (2014), p. 27 (cited on page 13).

[19] Paul Debevec. 'The light stages and their applications to photoreal digital actors'. In: *SIGGRAPH Asia* 2.4 (2012) (cited on page 15).

[20] Vincent Masselus, Philip Dutré, and Frederik Anrys. 'The free-form light stage'. In: *ACM SIGGRAPH 2002 conference abstracts and applications on - SIGGRAPH 02* (2002). DOI: 10.1145/1242073.1242275 (cited on page 15).

[21] Julian Michael Urbach et al. *Portable mobile light stage*. US Patent 9,609,284. Mar. 2017 (cited on page 16).

[22] Imari Sato et al. 'Using extended light sources for modeling object appearance under varying illumination'. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 1. IEEE. 2005, pp. 325–332.

[23] Yoav Y Schechner, Shree K Nayar, and Peter N Belhumeur. 'A theory of multiplexed illumination'. In: *null*. IEEE. 2003, p. 808 (cited on page 16).

[24] Taihua Wang. *Multispectral Light Stage for Object Classification*. 2019 (cited on page 16).

[25] Stefanos Zafeiriou et al. 'Face recognition and verification using photometric stereo: The photoface database and a comprehensive evaluation'. In: *IEEE transactions on information forensics and security* 8.1 (2012), pp. 121–135 (cited on page 16).

[26] Shaohua Kevin Zhou et al. 'Appearance characterization of linear lambertian objects, generalized photometric stereo, and illumination-invariant face recognition'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.2 (2007), pp. 230–245 (cited on page 16).

[27] RJ Woodham, Y Iwahori, and Rob A Barman. *Photometric stereo: Lambertian reflectance and light sources with unknown direction and strength*. University of British Columbia, Department of Computer Science, 1991 (cited on page 17).

[28] Daisuke Miyazaki and Katsushi Ikeuchi. 'Photometric stereo under unknown light sources using robust SVD with missing data'. In: *2010 IEEE International Conference on Image Processing*. IEEE. 2010, pp. 4057–4060 (cited on page 17).

[29] Roberto Mecca et al. 'Near field photometric stereo with point light sources'. In: *SIAM Journal on Imaging Sciences* 7.4 (2014), pp. 2732–2770 (cited on page 17).

[30] James J Clark. ‘Photometric stereo using LCD displays’. In: *Image and Vision Computing* 28.4 (2010), pp. 704–714.

[31] Nathan Funk and Yee-Hong Yang. ‘Using a raster display for photometric stereo’. In: *Fourth Canadian Conference on Computer and Robot Vision (CRV’07)*. IEEE. 2007, pp. 201–207 (cited on pages 17, 18).

[32] Lina Bi, Zhan Song, and Linmin Xie. ‘A novel LCD based photometric stereo method’. In: *2014 4th IEEE International Conference on Information Science and Technology*. IEEE. 2014, pp. 611–614 (cited on page 18).

[33] Anil K Jain and Stan Z Li. *Handbook of face recognition*. Vol. 1. Springer, 2011 (cited on pages 19, 20).

[34] Zhihong Pan et al. ‘Face recognition in hyperspectral images’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.12 (2003), pp. 1552–1560 (cited on page 20).

[35] Android Authority. *Facial recognition technology explained*. https://www.androidauthority.com/facial-recognition-technology-explained-800421/. 2019, Feb (cited on page 20).

[36] Keyurkumar Patel, Hu Han, and Anil K. Jain. ‘Secure Face Unlock: Spoof Detection on Smartphones’. In: *IEEE Transactions on Information Forensics and Security* 11.10 (2016), pp. 2268–2283. DOI: 10.1109/tifs.2016.2578288 (cited on page 20).

[37] Shaxun Chen, Amit Pande, and Prasant Mohapatra. ‘Sensor-assisted facial recognition’. In: *Proceedings of the 12th annual international conference on Mobile systems, applications, and services - MobiSys 14* (2014). DOI: 10.1145/2594368.2594373 (cited on page 20).

[38] Pocket-lint. *What is Apple Face ID and how does it work?* Sept. 2019 (cited on page 20).

[39] N. Erdogmus and S. Marcel. ‘Spoofing 2D face recognition systems with 3D masks’. In: *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*. 2013, pp. 1–8 (cited on page 21).

[40] G. Pan et al. ‘Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera’. In: *2007 IEEE 11th International Conference on Computer Vision*. 2007, pp. 1–8 (cited on page 21).

[41] Stephanie AC Schuckers. ‘Spoofing and anti-spoofing measures’. In: *Information Security technical report* 7.4 (2002), pp. 56–62.

[42] Yoav Y Schechner, Shree K Nayar, and Peter N Belhumeur. ‘Multiplexing for optimal lighting’. In: *IEEE Transactions on pattern analysis and machine intelligence* 29.8 (2007), pp. 1339–1354 (cited on page 35).

[43] Netanel Ratner and Yoav Y Schechner. ‘Illumination multiplexing within fundamental limits’. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–8 (cited on page 35).

[44] Alexander Sergeev and Mike Del Balso. ‘Horovod: fast and easy distributed deep learning in TensorFlow’. In: *arXiv preprint arXiv:1802.05799* (2018).

[45] Rattapoom Waranusast, Pongsakorn Intayod, and Donlaya Makhod. ‘Egg size classification on Android mobile devices using image processing and machine learning’. In: *2016 Fifth ICT International Student Project Conference (ICT-ISPC)*. IEEE. 2016, pp. 170–173.