Department of Human Biology

Division of Biomedical Engineering

A structured light solution for detecting scapular dyskinesis

Author: **Jaco Verster**

Supervisor: **Dr Janine Gray**

Co-supervisors: **Dr Sudesh Sivarasu**

Dr Tinashe Mutsvangwa

Submitted in fulfilment of the requirements for the degree of

MSc (Med) in Biomedical Engineering by dissertation

University of Cape Town

February 2018

Declaration

I, Jacobus Johannes Verster, hereby declare that the work on which this dissertation/thesis is based is my original work (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university.

I empower the university to reproduce for the purpose of research either the whole or any portion of the contents in any manner whatsoever.

Signature: Signed by candidate

Date: 2018-07-24

Abstract

Scapular dyskinesis is a common occurrence in overhead athletes, i.e. athletes who participate in any sport where the upper arm and shoulder is used above the athlete's head. However, no consensus has been reached on how to evaluate scapular dyskinesis quantitatively. In this thesis, we developed a measuring tool that can be used to evaluate certain key clinical parameters specific to scapular dyskinesis. The tool employs a 3D structured light computer vision approach to create a surface map of the soft-tissue across the scapula. This surface map is then analysed using surface curvature analysis techniques to identify the key clinical parameters associated with scapular dyskinesis. The main advantage of this method is that it provides a measurement tool that may facilitate future quantitative analysis of these key parameters. This may aid with diagnosis and monitoring of the condition by allowing measurement data to be collected both before and after treatment and rehabilitation. We expect that this tool will make the monitoring of treatment effectiveness easier while contributing to diagnostic computer vision.

Acknowledgements

I would like to thank the following people for their invaluable support throughout this project. Dr Janine Gray, my primary supervisor, for all your invaluable inputs and patience in guiding me. Without your expertise and knowledge this project would not have been possible. Dr Sudesh Sivarasu and Dr Tinashe Mutsvangwa, my co-supervisors, for their technical advice and general support in making this project a success. Mia Verster, my wife, for all your support and dedication to making my lifelong dream a reality. My brother, Eben Verster, for your endless support and hospitality. Finally, to Hanan Verster, my little boy, who was born during this undertaking.

Table of Contents

List of Figures

List of Tables

List of abbreviations

AA	Acromion angle
AMC	Acromion marker cluster
BMI	Body Mass Index
CCD	Charge-coupled device
CT	Computer tomography
H-W	Hip-to-waist
IA	Inferior angle
MRI	Magnetic resonance imaging
MB	Medial border
NLMF	Non-local means filter
RoM	Range of motion
RGB	Red, green and blue
RMS	Root-mean-square
SDT	Scapular Dyskinesis Test
STAs	Soft tissue artefacts
SSM	Statistical shape model
SA	Superior angle
SC	Surface curvature
3D	Three dimensional
2D	Two dimensional
WHO	World Health Organisation

1 Introduction

1.1 Background

The shoulder is one of the most complex joints in the body and a common site for injuries. Injuries are frequently associated with repetitive overhead motion of the arm during sporting, recreational or job-related activities and can lead to functional impairment and significant healthcare costs (Brooks, 2006). When the shoulder is studied from a biomechanical perspective, the scapula, and specifically the coordination between the scapula and the humerus, plays a critical role in normal shoulder function (Kibler, 1998). Abnormal movement or resting position of the scapula is defined as scapular dyskinesis and it has been associated with shoulder pain (Kibler & McMullen, 2003). Even though the exact cause-and-effect relationship between scapular dysfunction and specific pathologies or injury mechanisms are not yet fully understood, there is enough evidence to support the clinical importance of the scapula to prompt further investigation into the nature of involvement of the scapula in shoulder pain (Kibler et al., 2013). Furthermore, it has been shown that the measurement of shoulder and scapula kinematics provides relevant information for the diagnosis and treatment of clinical disorders, rehabilitation techniques, sports performance and injury prevention (Lempereur, Brochard, Leboeuf, & Rémy-Néris, 2014).

1.2 Clinical problem

In order to investigate the relationship between scapular dysfunction and specific pathologies or injury mechanisms further, it is essential to have an objective, reliable and accurate method to assess scapular dyskinesis in a clinical setting. Clinical assessment of the scapula is, however, inherently challenging due to the three-dimensional (3D) nature of dyskinesis and the large amount of soft-tissue involvement. The surrounding muscle (both attached to and overlying the scapula) and the large amount of under-skin movement associated with the scapula makes the direct measurement of bony landmarks difficult. Several methods for measuring scapular movement have been proposed, for example bone pins, scapula locators, bi-plane fluoroscopy, reflective skin-markers and electromagnetic trackers. Even though many of these methods have shown adequate levels of reliability in a clinical environment, they

present disadvantages that often makes them impractical to use in a clinical setting or clinical research. Some of these disadvantages include being highly invasive, radiation exposure, too time consuming, limited to two dimensions or static positions only, or expensive to set up.

In clinical practice, assessment of the scapula is mostly done through visual observations (the scapular dyskinesis test (McClure, Tate, Kareha, Irwin, & Zlupko, 2009)) and it is proposed that this is partly due to the disadvantages of the current measurement systems. Although the visual test has shown sufficient reliability for clinical use, it is limited to only being able to identify the presence of scapular dyskinesis but not to accurately quantify it.

1.3 Motivation

An objective measurement system, that can identify and quantify scapular dyskinesis would be valuable in a clinical setting and would allow clinicians to better direct interventions aimed at improving the kinematics of the scapula in those who suffer from shoulder pain. In addition, it would allow the effectiveness of rehabilitation programmes aimed at improving dyskinesis to be monitored. It will also allow researchers to investigate the cause-and-effect relationship between scapular movement and dysfunction as well as gain insight into which symptoms are causative or compensatory in nature.

1.4 Research approach

This research project includes both clinical and engineering aspects. A clinical problem was identified (measuring scapular dyskinesis) and an engineering design approach was followed to develop an appropriate solution to the problem. The developed solution was tested in a clinical setting using a single group of participants to determine its effectiveness. The research was novel and explorative in nature.

1.5 Hypothesis, aim and objectives

1.5.1 Hypothesis

We propose that a reliable, quantitative measurement tool for scapular dyskinesis would allow us to investigate and better understand the cause-and-effect relationship between

shoulder dysfunction and scapular dyskinesis. Such a system would allow for better informed diagnoses and potentially facilitate further research opportunities to study the scapula.

1.5.2 Aim

The aim of this study is to design an accurate scapular tracking system to measure scapular dyskinesis, build a proof of concept prototype and test it in a clinical setting.

1.5.3 Objectives

The following objectives have been identified:

1. To evaluate current clinical methods of diagnosing scapular dyskinesis and select key clinical parameters to measure.
2. To design a system (hardware and software) for accurately measuring the key clinical parameters to assist with scapular dyskinesis diagnosis.
3. To test and verify the suitability of the system for measuring the key parameters.
4. To test the system in a clinical setting to verify its ability to measure scapular dyskinesis.

1.6 Dissertation overview

This document describes the research, clinical and engineering methodologies used in this study. Chapter 2 (Literature Review and Theory) provides the anatomical and clinical background for scapular dyskinesis and the theory that supports the engineering solution. Chapter 3 (Methodology) is divided into two sections. Section I details the inputs, detailed design and laboratory testing of the engineering solution while Section II provides the clinical rationale for the solution as well as the details of the clinical study. Chapter 4 (Results) includes the final results of the clinical testing and an in-depth discussion about the outcomes. This document concludes with Chapter 5 (Conclusion and Recommendations) where the major conclusions drawn from the study are discussed followed by a brief mention of other potential applications and recommended future work.

2 Literature Review and Theory

2.1 The shoulder girdle

The shoulder girdle connects the upper limb to the axial skeleton and consist of two bones, the scapula and clavicle (Figure 1). The girdle is connected to the anterior rib cage at the sternoclavicular joint. In addition to this anatomical joint, there is a physiological joint (or muscular connection) that connects the scapula to the posterior rib cage, called the scapulothoracic joint. This muscular connection allows the scapula to glide over the rib cage and enable the shoulder girdle to move through a large range of motion (RoM). The shoulder girdle has two other joints where the clavicle connects to the scapula (acromioclavicular joint) and where the humerus connects to the scapula (glenohumeral joint).

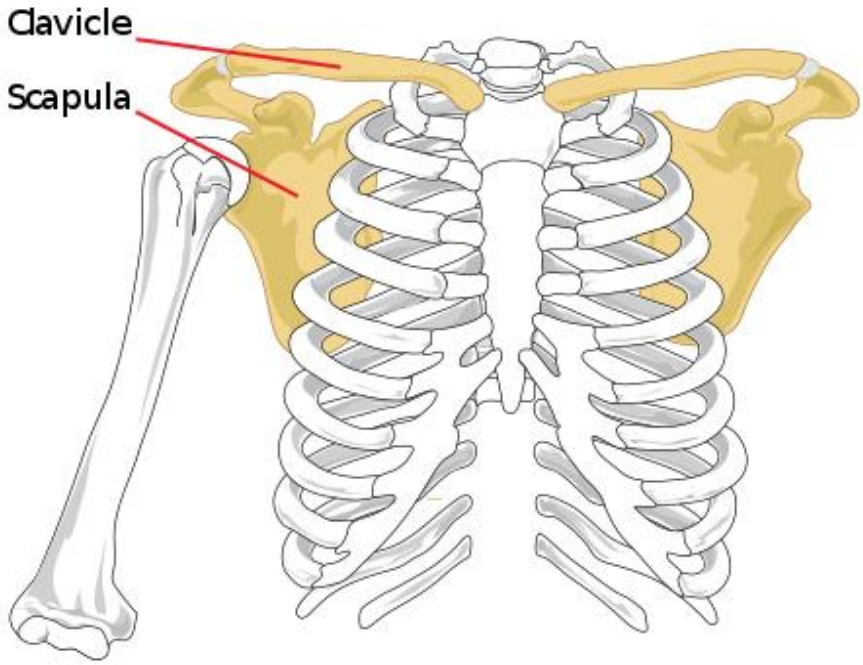


Figure 1: The shoulder girdle (Wikipedia, 2007).

2.2 The scapula

The scapula, also known as the shoulder blade, is a flat triangular bone with very distinct features. The most notable physical landmarks of the scapula are the inferior angle, medial border, spine and acromion (Figure 2). These landmarks are easy to palpate (examine by touch) and are often visibly identifiable by the skin on the back.

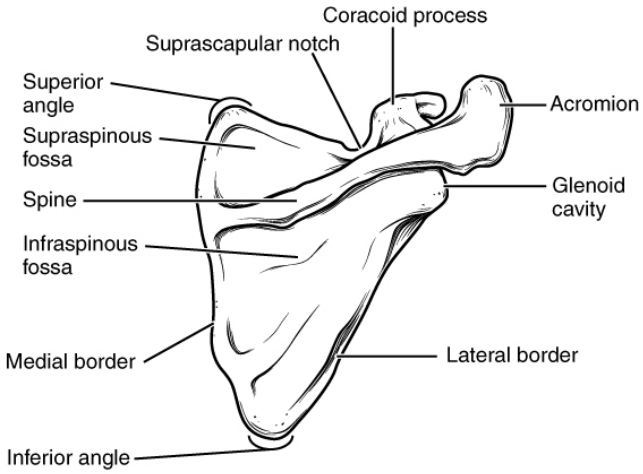


Figure 2: Bony features of the scapula (OpenStax, 2013).

The scapula serves as an attachment site for many of the shoulder girdle muscles (Figure 3). These muscles include the main stabilisers of the glenohumeral joint, commonly referred to as the rotator cuff, namely the supraspinatus, infraspinatus, teres minor and subscapularis. The medial border, inferior angle and superior angle of the scapula are only covered by the very thin trapezius muscle, while the spine and acromion have no musculature covering them. This permits easy palpation of these landmarks and a partially visible scapula in people with a lean body composition.

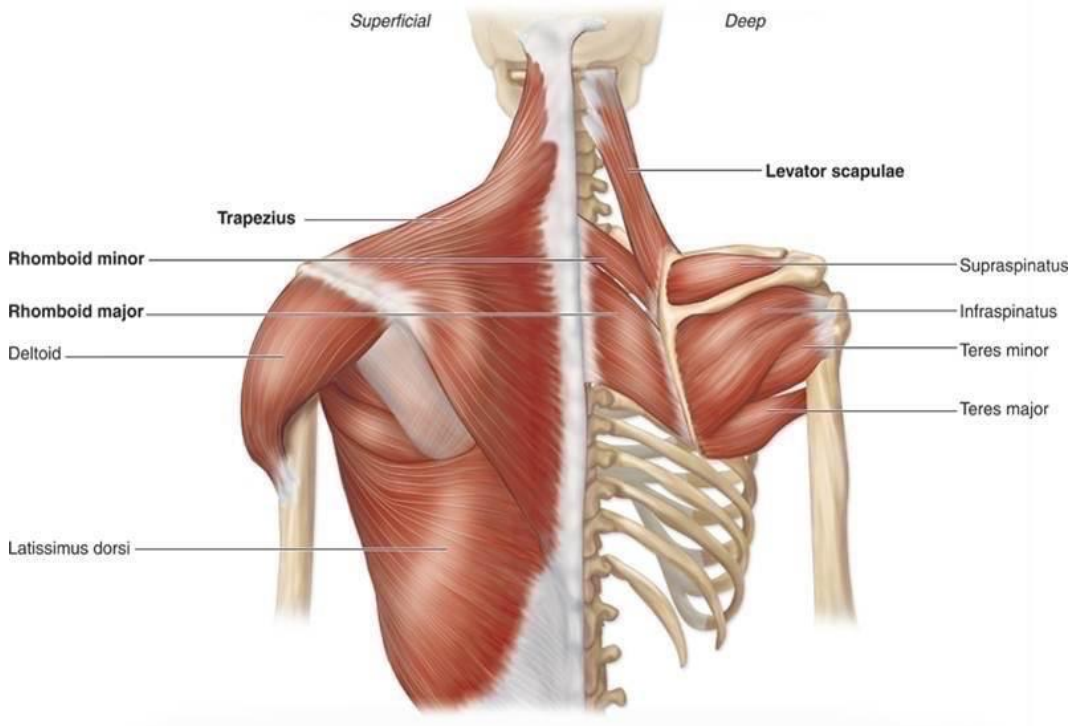


Figure 3: The superficial (left) and deep (right) muscles of the shoulder girdle (Studyblue, 2013).

2.2.1 Movements of the scapula

The muscles that attach the scapula to the rib cage not only keep it in position, but also enable it to glide and rotate over the rib cage. This makes the shoulder girdle extremely mobile. The movements of the scapula are organised around a 3D axis embedded in the scapula and include elevation and depression, abduction and adduction, upward and downward rotation as well as internal and external rotation (Figure 4). Movement descriptions and typical RoM for the scapulothoracic joint are shown in Figure 4 and Table 1.

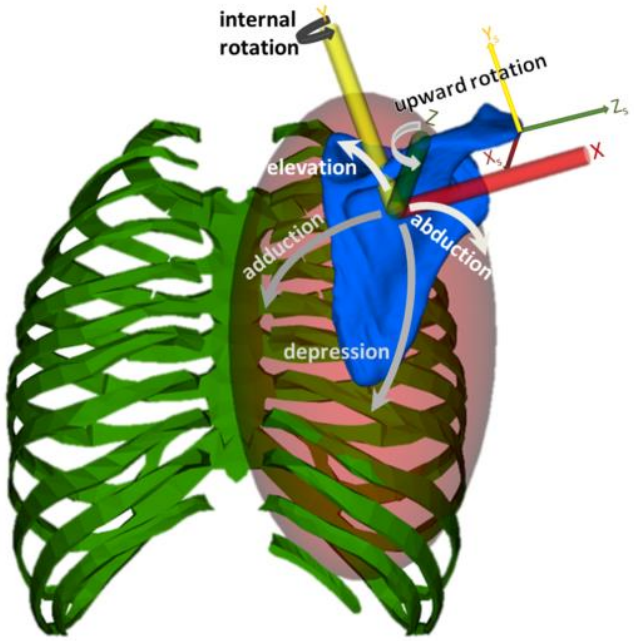


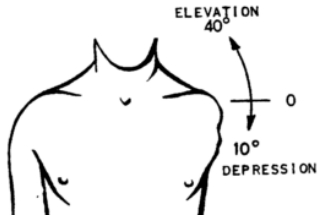
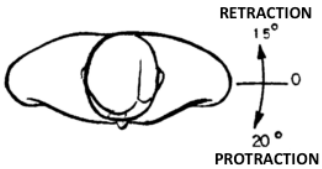
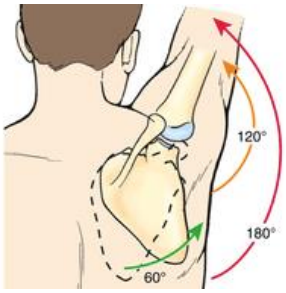
Figure 4: Movements of the scapula: Elevation (up) and Depression (down), Abduction (right) and Adduction (left), Upward/Downward rotation and Internal/External rotation (Seth, Matias, Veloso, & Delp, 2015).

2.2.2 Scapular biomechanics

The first role of the shoulder girdle is to move the glenohumeral joint into various positions to enable the shoulder and arm to be used for daily as well as athletic activities. It allows for the ball-and-socket configuration of the glenohumeral joint to be maintained throughout the full RoM. The second is to provide a stable base for glenohumeral articulation. These two roles are often fulfilled at the same time as the arm is used to generate, absorb, or transfer forces that accomplish work or athletic tasks. The bony anatomy of the shoulder girdle provides the structural foundation from which the various shoulder muscles act to generate the forces responsible for glenohumeral stability and mobility. The scapula is central to this foundation and therefore plays a critical role in optimal shoulder function.

The muscles of the scapulothoracic joint work in a coordinated manner to dynamically position and stabilise the scapula for optimal glenohumeral mechanics. When the scapula is performing as it should, its movements will be observed as smooth and coordinated and the whole scapula will sit tightly on the rib cage during both rest and movement.

Table 1: Scapulothoracic joint movement and RoM.

Motion	Description	Typical RoM
Elevation	Upward movement of the scapula.	 Image reference: (John Michael, 2008)
Depression	Downward movement of the scapula to return to resting position.	
Adduction (retraction)	Scapula moves medially towards the spine.	 Image reference: (John Michael, 2008)
Abduction (protraction)	Scapula moves laterally away from the spine.	
Upward rotation	Inferior angle of the scapula moves superolaterally to follow the arm as it is lifted up: when the arm is abducted 180°, 60° occurs by rotation of the scapula and 120° by rotation of the humerus.	 Image reference: (Best Performance Group, 2014)
Downward rotation	Inferior angle of the scapula moves inferomedially back to resting position.	

2.2.3 Scapular dyskinesis

Any weakness or dysfunction in scapula musculature can disrupt the delicate coordination required to effectively position and/or move the scapula and result in suboptimal joint mechanics. When the scapula can no longer perform as required, shoulder position, motion, stability, muscle performance and motor control become ineffective (Kibler et al., 2013). This, in turn, may predispose an individual to glenohumeral joint injuries (Paine & Voight, 2013). The alterations in normal resting position and motion of the scapula is defined as scapular

dyskinesis (Kibler et al., 2013). Clinically, the term scapular dyskinesis is used to describe the general loss of control of scapular motion and position.

There are many factors that contribute to scapular dyskinesis, which include amongst others, muscle imbalances or tightness, nerve damage and/or joint instability. Scapular dyskinesis may be the result of an injury or may aggravate an existing injury. Although several contributing factors have been identified, the exact cause-and-effect relationship is not yet fully understood (Kibler et al., 2013). It is, however, clear that there is substantial evidence for the presence of scapular dyskinesis in patients with shoulder pain (Kibler et al., 2013).

As a result of the strong link between scapular dyskinesis and shoulder pain, many different clinical assessment methods for precisely evaluating scapular dyskinesis have been proposed. Currently, the Scapular Dyskinesis Test (SDT), as described by Kibler *et al*. in a statement from the 2013 Scapular Summit, seems to be the gold standard for evaluating scapular dyskinesis (Kibler et al., 2013). Other methods, like the SICK scapula syndrome used by Burkhart *et al*. to categorise altered scapular kinematics into three classes or types (Burkhart, Morgan, & Kibler, 2003), also exist but are not yet seen as gold standard.

In summary, the SDT involves, amongst other things, the clinical observation of the following:

- Inferior and medial borders of the scapula for prominence or winging during rest (Figure 5).

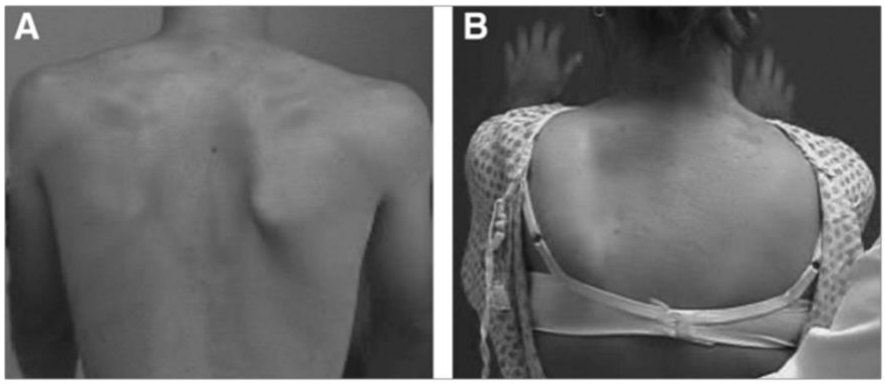


Figure 5: A, prominent right inferior medial scapular border (winging) and B, entire left medial border prominence (Uhl, Kibler, Gecewich, & Tripp, 2009).

- Lack of smooth coordinated movement of the scapulae during arm flexion and abduction as demonstrated by early scapular elevation during ascending arm forward flexion, or rapid downward rotation during arm lowering (Figure 6).

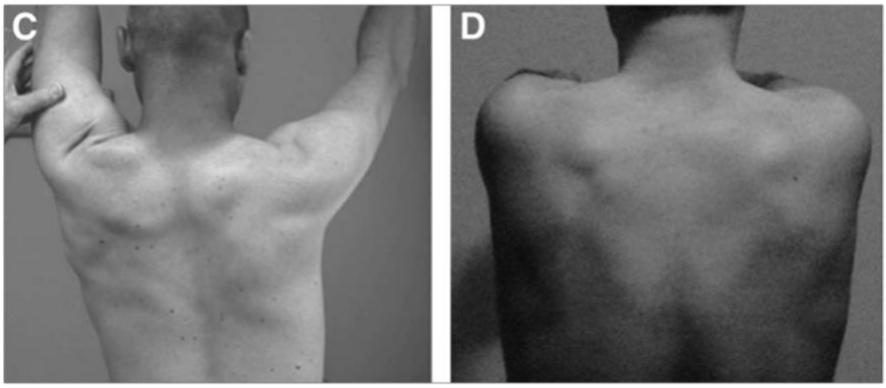


Figure 6: C, Early left scapular elevation during arm motion and D, normal symmetric scapular motion (Uhl et al., 2009).

During an evaluation, a patient is typically observed from the back with the arms at rest, and during flexion and abduction of the arms to detect an abnormal or asymmetric positioning or motion of the scapula. The tests conclude with a simple “yes” or “no” classification for the presence of scapular dyskinesis with some description of winging (medial border prominence) or dysrhythmia (premature, sporadic or excessive motion of the scapula during elevation or lowering of the arm). The yes/no method is a good screening tool for the presence of scapular dyskinesis but provides no quantitative measurements.

2.3 Methods of evaluating the scapula

To address the limitations of the current clinical evaluation methods for scapular dyskinesis, which is predominantly qualitative, the aim of this study was to develop a quantitative measurement tool. A tool that can measure the position of the scapula relative to another point of reference, also known as scapular kinematics.

Several attempts have been made to study scapular kinematics, but all of these methods have several limitations and difficulties that have not been resolved yet. Two-dimensional (2D) image and video analysis methods are commonly used but do not produce quantifiable data and have the same drawbacks as visual observation (McClure et al., 2009). Cadaver studies provide highly accurate joint position measurements, but are unable to accurately duplicate the complex muscle forces associated with shoulder movements in living subjects (Bey, Zauel,

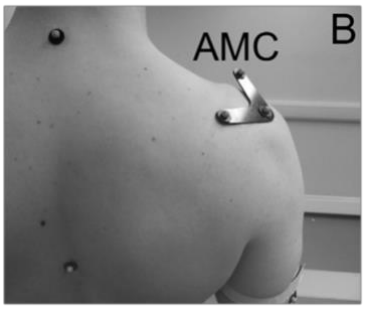
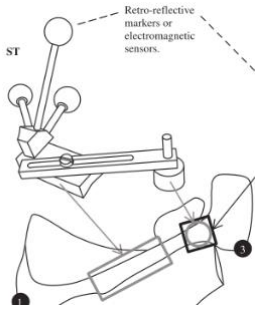
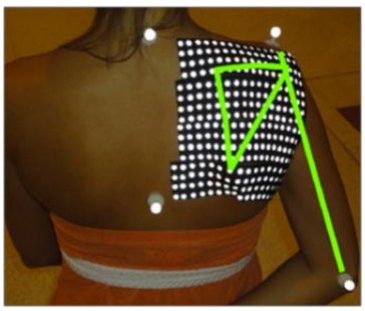
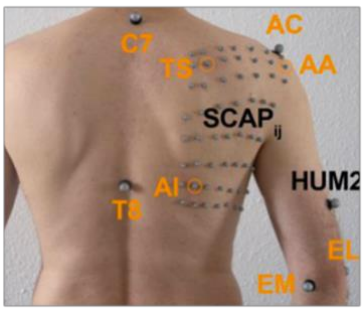
Brock, & Tashman, 2006). Three-dimensional (3D) wing computer tomography (CT) and magnetic resonance imaging (MRI) provide highly accurate measurements, but only allow for static measurements, which are limited to body positions that can be accommodated by the size of the scanner (Park et al., 2013). Cutaneous (skin) marker based methods are the least invasive and most convenient, but lack accuracy due to soft-tissue interference (Matsui, Shimada, & Andrew, 2006). Bone pins provide pin-point scapula location accuracy during some movements, but involve invasive surgical insertion of the pins, which are subject to soft-tissue effects that alter and limit movements (McClure, Michener, Sennett, & Karduna, 2001).

Of all these methods, X-ray fluoroscopy is considered to be the gold standard. It provides highly accurate scapula and joint position measurements and can be used for both static and dynamic measurements. This method uses two simultaneous X-ray fluoroscopic images to determine the position of the scapula. Although less invasive than bone pins, the drawback of this method is unwanted radiation exposure, expensive equipment requirements and relatively limited arm movement. Charbonnier *et al.* used fluoroscopy to assess the effectiveness of their patient-specific shoulder joint kinematics measurement technique for studying glenohumeral translations (Charbonnier, Chague, Kolo, Chow, & Ladermann, 2014).

2.3.1 Marker based methods

For reasons involving costs, practicality and non-invasiveness, the most common methods for measuring scapular kinematics in a laboratory setting are cutaneous marker based systems. These methods make use of either electromagnetic or optoelectronic markers that are placed on bony landmarks of the scapula. A multi-camera optical motion tracking system is then used to determine the position of the markers in 3D space by means of complex triangulation. The main challenge faced by marker based techniques, however, are the inaccuracies introduced to the final measurements due to manual marker placements and soft-tissue interferences. For markers placed on the medial border of the scapula, errors of up to 87mm have been recorded (Matsui et al., 2006).

Table 2: Summary of the most common marker based methods.

Method	Example	Description	Limitations
Acromion marker cluster (AMC)		The acromion marker cluster consists of three reflective markers fitted to a structure that is then fixed to the skin on the flat portion of the acromion. Image: Warner *et al.* (2012).	Provides no information about the medial border or inferior angle of the scapula.
Scapula tracker		The scapula tracker is similar to the AMC, the only difference being that it is fixed to the skin on the spine of the scapula. Image: Prinold *et al.* (2011).	Provides no information about the medial border or inferior angle of the scapula.
Scapular locator		The scapula locator usually consists of three locating pins, calibrated to the patient's scapular landmarks. It is placed on the scapula by manual palpation. Image: Brochard *et al.* (2011).	Very time consuming and does not allow for dynamic measurements.
Surface mapping with markers		Mattson *et al.* used Kinesio tape with 300 markers that create a surface map. A scapula template was fitted to the surface map to estimate the scapular position. Image: Mattson *et al.* (2012).	The degree to which Kinesio tape affects the scapula movements and proprioception is not known. Very time consuming.
Multiple markers on scapula (surface mapping)		Charbonnier *et al.* used multiple markers on the skin of the scapula. A patient specific MRI was used in combination with optical marker measurements to determine scapular position. Image: Charbonnier *et al.* (2014).	Marker placement is very time consuming. An MRI scan is required for each patient.

In addition to the measurement inaccuracies, marker methods require a time-intensive marker placement and calibration procedure for each patient. Manual marker placements are required for every test session, negatively affecting the repeatability of such a system because the markers can never be placed on exactly the same locations as before.

In a systematic review of the validity and reliability of 3D marker based methods, Lempereur *et al.* (2014) identified six different methods that are most commonly used. Methods varied based on marker locations, calibration strategies and data computation algorithms. Based on the review, Lempereur *et al.* (2014) recommended the acromion marker cluster (AMC) method because of its accuracy. The main marker-based methods are briefly described in Table 2.

2.3.2 Surface mapping methods

Even though the most common methods for measuring scapular kinematics are still cutaneous marker based, soft-tissue interference remains a challenge. In their review, Lempereur *et al*. (2014) made several recommendations aimed at future research initiatives and concluded in saying that it is highly probable that a marker-less approach would be the most accurate solution. Further research into marker-less methods is therefore required to find ways to overcome the soft-tissue interference problem.

Mattson *et al*. (2012)(Table 2), took a step in this direction by using surface markers to create a 3D surface map of the skin across the scapula. They used complex computational algorithms to finally approximate the position of the scapula from the 3D surface map. They reported that the Kinesio tape may have altered the position of the scapula and that the tape proved unreliable in keeping the markers in the correct position due to tearing.

Moiré fringe projection technique

In another effort to avoid using marker based methods, Gomes *et al.* (2010) describes a method for measuring scapular kinematics using Moiré topography. Moiré topography uses an interference pattern, called a Moiré pattern, to create an accurate 3D reconstruction of the object under study. An interference pattern is an optical occurrence created when one set

of curves is positioned at a small offset distance or angle over another set that is identical to the first (Oster, 1988). In practice, this is achieved by either using a physical screen and some illumination to cast a shadow (Figure 7), or by projecting a pattern onto the surface and using software to compare the captured image with the original pattern. Moiré topography techniques have been successfully used to study scoliosis (Porto, Gurgel, Russomano, & Farinatti, 2010). A Moiré topography system is shown in Figure 7.

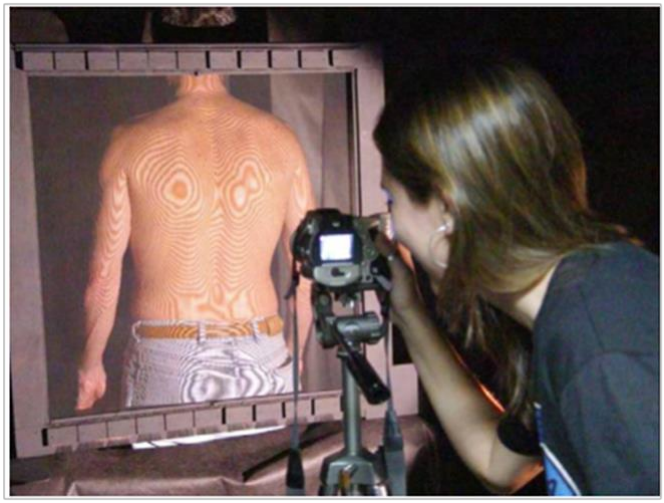


Figure 7: Moiré topography scan of the human back (Porto et al., 2010).

In the study by Gomes *et al.*, a 3D surface map of the skin over the scapula is obtained using Moiré topography and once the surface data was obtained, a computer algorithm was used to approximate the subcutaneous position of the scapula. They were able to make quasi-static 3D scapular measurements using the technique but reported that excessive soft-tissue may distort the measurements. Some advantages of Moiré topography include being non-invasive, no radiation exposure, and fast evaluation of many patients in a relatively low-cost manner.

Structured light 3D surface imaging

Another surface mapping technique, that makes use of optical techniques, is called structured light 3D surface imaging. This technique is commonly used in computer vision applications to reconstruct objects and was successfully used by Thabet *et al*. (2014) to study the human foot in motion (Figure 8). Conventional passive 3D stereo vision techniques struggle to reconstruct the human body due to a lack of surface texture. In passive stereovision systems (human eyes, for example) surface texture is used to identify differences between two images, captured from two different viewing angles. These differences are used to calculate depth of the scene.

Structured light is seen as an active stereovision solution where artificial texture is created on a smooth surface by projecting a light pattern onto the surface (L. Zhang, Curless, & Seitz, 2002). The projected light pattern, in combination with a calibrated projector-camera setup, can be used to reconstruct the target surface using optical triangulation.

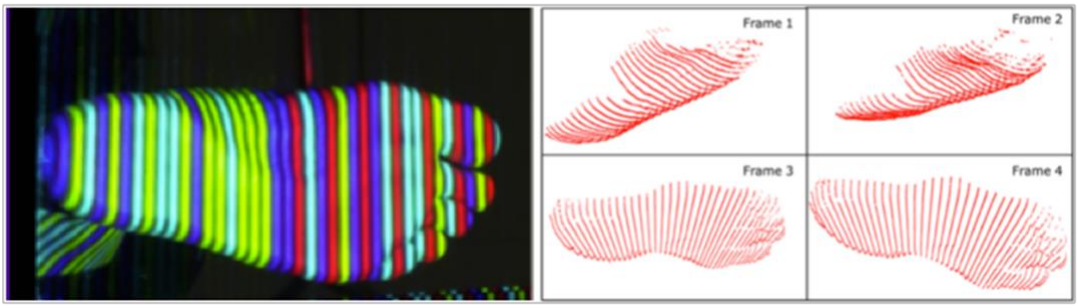


Figure 8: Left - the structured light pattern used to illuminate the foot surface. Right - examples of the 3D reconstructions (Thabet et al., 2014).

Similar to Moiré topography methods, structure-light approaches are low-cost, non-invasive, without radiation exposure and allow for rapid shape acquisition. Even though no evidence of a structured light evaluation method for scapular dyskinesis could be found in literature, the technique is similar to Moiré topography and would be well suited for creating a 3D surface map of the skin over the scapula. Some of the limitations of structured light methods include reconstruction errors due to surface texture and colour variation, the potential need for a completely stationary object to avoid movement errors and a sensitivity to the initial calibration procedure. These limits can, however, be overcome or avoided by selecting the correct decoding pattern and by carefully following an established calibration protocol.

Studies involving structured light are commonly found in the computer vision field, which is very active, and tend to focus heavily on the technical details of implementations (Eiríksson, Wilm, Pedersen, & Aanæs, 2016)(Drerup, 2014)(Geng, 2011). Moiré topography applications, on the other hand, are not as well described in literature and no technical descriptions of an implementation could be found. Any further research into developing a 3D surface map evaluation method for scapular dyskinesis would therefore be easier to implement using a structured light approach.

2.4 Structured light theory

A structured light imaging approach was chosen for this study. The rationale behind this decision as well as the detailed design of the system is discussed in the next chapter. Before the design of the system is discussed, however, some background knowledge on structured light systems, camera theory and optical calibration is required.

In principle, structured light 3D surface imaging makes use of a specifically designed 2D light pattern to illuminate an object surface to generate artificial texture. An imaging sensor is then used to capture a 2D image of the object under structured illumination. If the object surface is completely flat without any 3D variation or rotation, the captured image will display the illumination pattern exactly as it was projected. However, when the object surface is nonplanar and contains 3D variation or rotation, the projected illumination pattern is distorted as seen from the view of the camera (Figure 9).

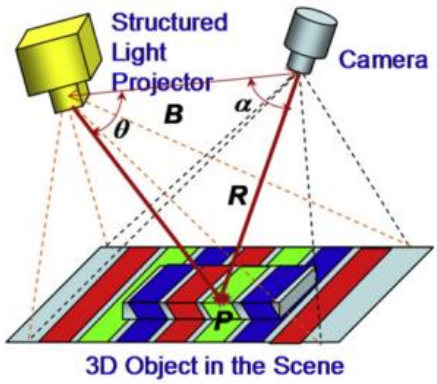


Figure 9: Structured light imaging working principle (Geng, 2011).

The 3D shape of the object surface can be extracted from the distortion information by comparing the distorted line (as seen by the camera) directly with the straight line (as seen by the projector) and calculating the x, y and z coordinates of several points on the 3D surface using triangulation. The triangulation principle (from basic trigonometry) that is used to calculate the 3D coordinates of point P (Figure 9) in relation to the camera (as conveyed by the vector R), is expressed as:

$$R = \frac{B \cdot sin(\theta)}{\sin(\alpha + \theta)} \tag{1}$$

In order to use this expression, two requirements must be met. Firstly, the point P on the object surface must be identified in both the 2D camera image and the 2D projected image.

In other words, the camera and projector image pixels must be in correspondence. The structure of the projected pattern is used to establish this correspondence. Secondly, the precise 3D locations for both the optical centres of the camera and the projector must be known.

2.4.1 Encoding strategies

Many different structured light imaging techniques exist, each using a different pattern design to facilitate correspondence. The pattern design is commonly referred to as the encoding strategy and the type of pattern dictates how many camera images are required to establish correspondence. Some strategies use a continuously varying pattern or distinct stripes while others use a grid or a pattern that is broken up into several images projected in sequence. Examples of several different encoding strategies are shown in Figure 10.

Each encoding strategy offers different advantages and disadvantages and choosing a strategy requires a performance trade-off between accuracy, resolution (number of points) and speed (number of images) (Eiríksson et al., 2016). For example, a grid pattern requires a single shot (one camera image) but will not produce a very dense point map because only the points on the grid intersections are in correspondence. On the other hand, a binary code pattern requires multiple shots but every pixel is in correspondence (Herakleous & Poullis, 2014).

Figure 11 shows an example of an encoding strategy where a deBruijn colour stripe pattern was used, using one camera image per reconstruction. Correspondence was established between the lines of pixels located on the edges of the colour stripes, i.e. where one colour changes to another.

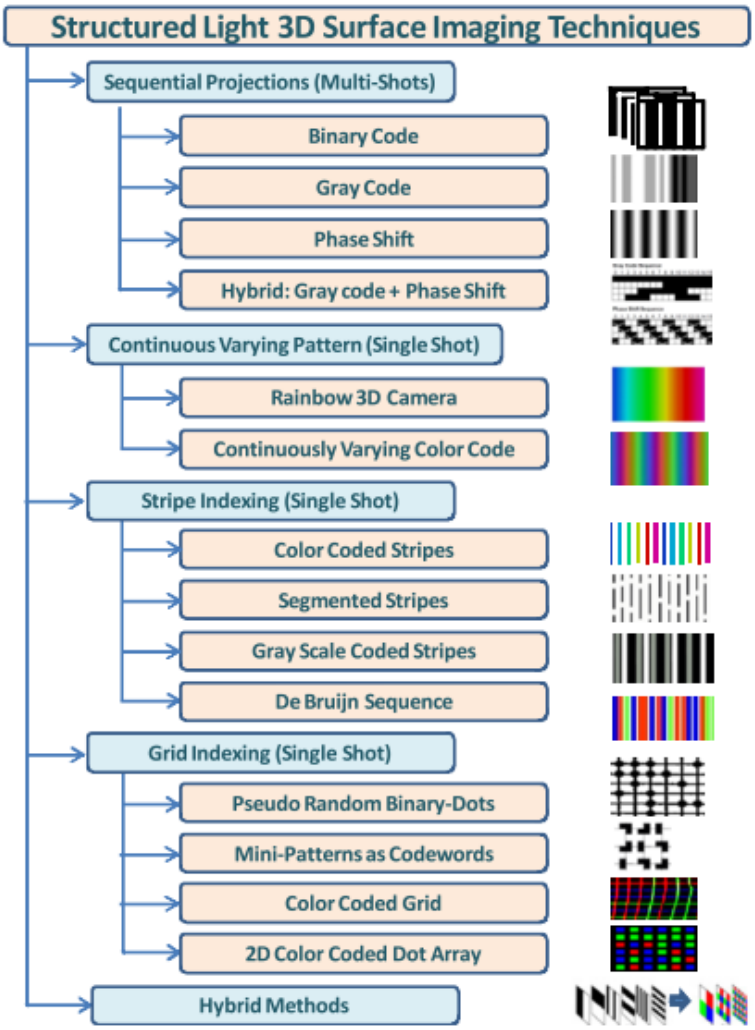


Figure 10: Classification of structured light imaging techniques (Geng, 2011).

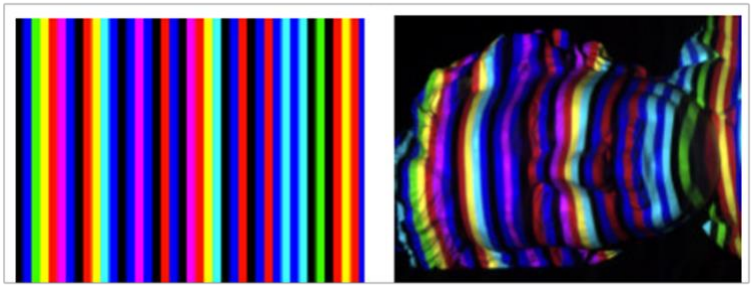


Figure 11: Example of a projector image (left) and a camera image (right) (L. Zhang et al., 2002).

2.4.2 Camera-projector calibration

In order to calculate the 3D location of a point on the surface object, the precise 3D locations (extrinsic parameters) as well as the optical properties (intrinsic parameters) for both the camera and the projector must be known. This is achieved by a camera-projector calibration

process. This process is critical as the accuracy of the final 3D reconstructions will depend on the calibration precision.

Many different camera-projector calibration models exist but for this project a well-known and widely respected method based on the work of Zhang *et al.* (2006) was used. This method is well documented and computerised implementations have been written in Matlab (Bouguet, 2015) and other programming languages (C++, Python and Java) (OpenCV, 2017).

The camera model

The pinhole model is most commonly used to describe the intrinsic and extrinsic parameters of a camera system. Intrinsic parameters include pixel size, pixel skew factor (height-to-width ratio), focal length and principle point. Extrinsic factors include the rotation and translation from the world coordinate system to the camera coordinate system (S. Zhang & Huang, 2006). Figure 12 illustrates a typical pinhole camera model where 'p' is an arbitrary point on an object with coordinates (x^w, y^w, z^w) and (x^c, y^c, z^c) in the world coordinate system $\{o^w; x^w, y^w, z^w\}$ and camera coordinate system $\{o^c; x^c, y^c, z^c\}$, respectively. The coordinate of its projection in the image plane $\{o; u, v\}$ is (u, v).

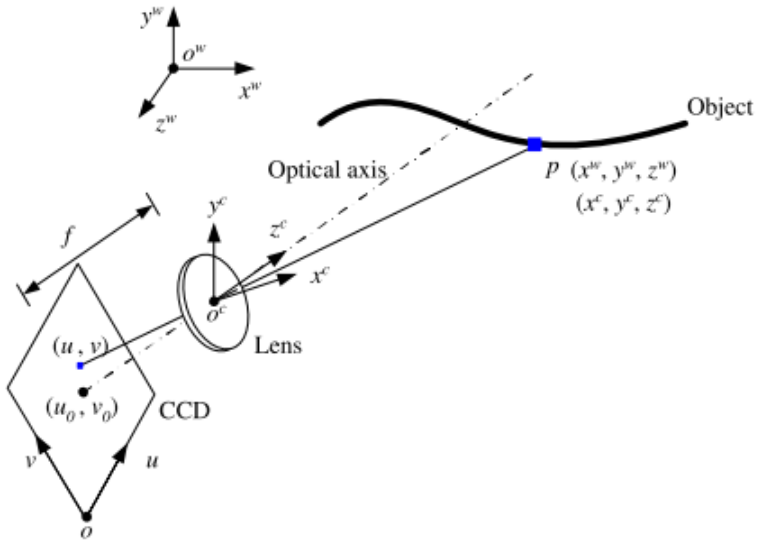


Figure 12: Pinhole camera model (S. Zhang & Huang, 2006).

The relationship between a point on the object and its projection on the image sensor can be described as follows based on a projective model:

$$sI = A[R, t]X^w, \tag{2}$$

where $I = \{u, v, 1\}^T$ is the homogenous coordinate of the image point in the image coordinate system, $X^w = \{x^w, y^w, z^w, 1\}^T$ is the homogeneous coordinate of the point in the world coordinate system, and 's' is a scale factor. $[R, t]$ is the extrinsic parameters matrix and represents rotation and translation between the camera coordinate system and world coordinate system. A is the camera intrinsic parameters matrix, expressed as:

$$A = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where (u_0, v_0) is the coordinate of principle point, f_u and f_v are the focal lengths along the u and v axes of the image plane, and γ is the parameter that describes the skewness of two image axes. Equation (2) represents the linear model of the camera. More complicated, non-linear camera models do exist, but a linear model is adequate for describing most systems.

For the purpose of this structured light application, the projector can be considered as an inverse camera where images are projected instead of captured. Therefore, the same camera model can be applied to the projector (in reverse) during calibration.

Camera calibration

The Camera Calibration Toolbox for Matlab by Bouguet (2015) provides an easy to use calibration tool based on the work of Zhang *et al.* (S. Zhang & Huang, 2006). This method uses multiple images of a printed flat checkerboard to obtain the intrinsic parameters of the camera. An example of a checkerboard image with Matlab corner detection is shown in Figure 13. A full explanation of the implementation in Matlab can be found in the toolbox documentation (Bouguet, 2015).

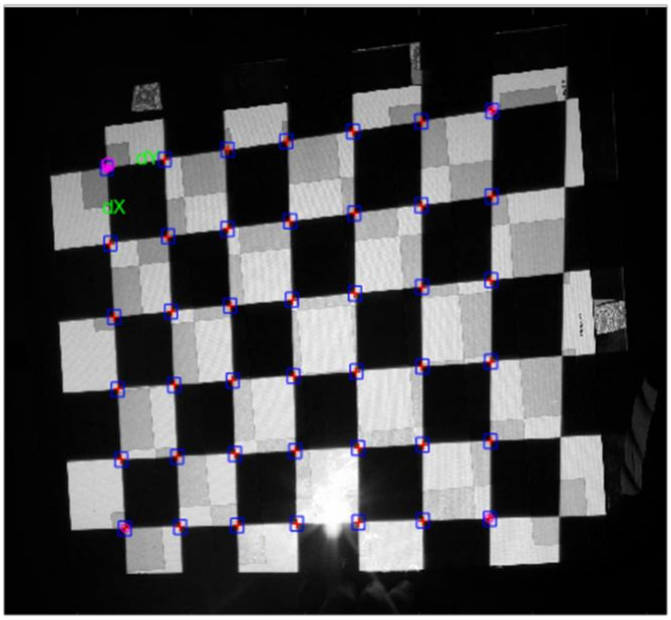


Figure 13: Example of a checkerboard pattern with corner detection shown.

Camera colour perception

Due to the importance of colour for this structured light application in this project, a short explanation of how cameras perceive colour is required. Modern digital cameras detect light by means of a charge-coupled device (CCD) or image sensor. It consists of a grid of light sensors that convert light impulses into electrical signals. In order to distinguish colours, most image sensors are covered with a blue, green and red colour grid called a Bayer colour filter (US3971065, 1976). The filter is arranged in such a way that every four pixels contains one blue, one red and two green segments. This exposes the image sensor to 25% red, 50% green and 25% blue light and allows the camera to store the red, green and blue (RGB) information in three separate colour channels. Bayer chose this arrangement to mimic the physiology of the human eye, which is more sensitive to green light. Figure 14 shows an ideal case Bayer colour filter and its working principle.

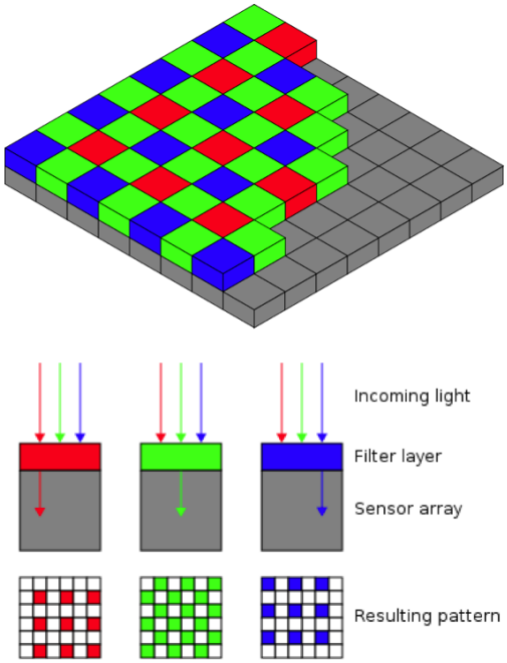


Figure 14: Bayer colour filter array (Wikipedia, 2006).

Projector calibration

The projector is unable to capture images like a camera, but once the intrinsic parameters of the camera are known, the camera can be used to capture images on behalf of the projector. In order to avoid errors from the camera calibration to influence the projector calibration, both systems need to be calibrated at the same time. Falcao *et al*. (2009) developed a Projector-Camera Calibration Toolbox, that serves as an extension to the Camera Calibration Toolbox by Bouguet, where this can be done using Matlab.

As with the camera calibration, a flat surface with printed checkerboard pattern is used but the same pattern is also projected onto the surface. Van Kessel proposed an elegant solution to minimize the size of the flat surface by using a blue printed pattern and a red projected pattern (Van Kessel, 2013). This allows both the printed and the projected pattern to be viewed on the same surface separately by looking at either the blue or the red colour channel of the camera. Figure 15 shows an example of the calibration patterns used for this project, based on Van Kessel's concept. Going back to Figure 13 above, notice how the red colour channel of the camera was used to highlight the red projected image and diminish the blue printed pattern.

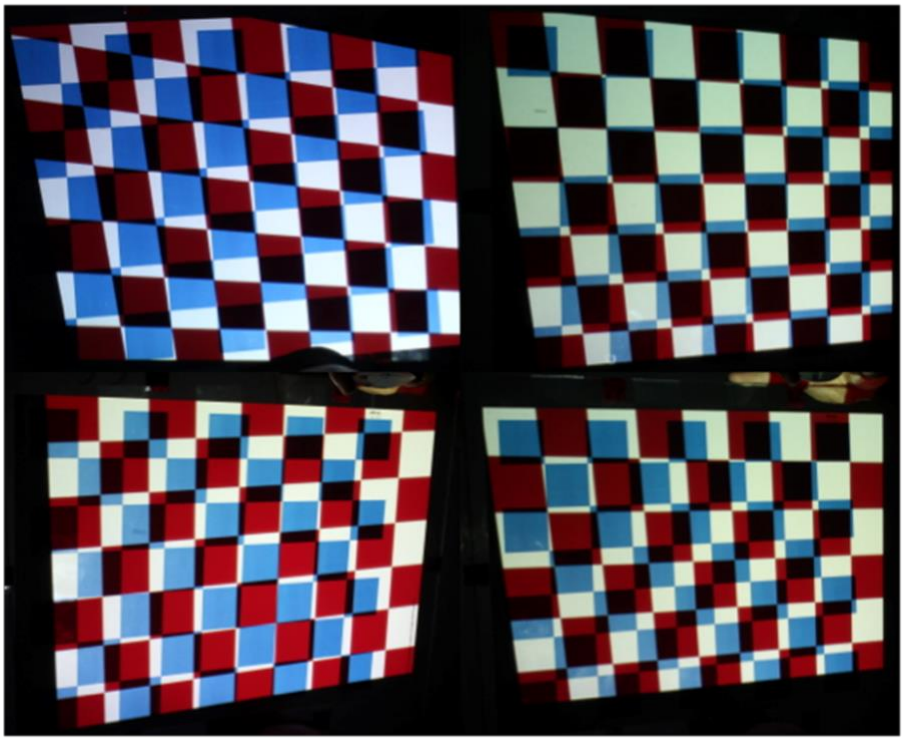


Figure 15: Red checkerboard pattern projected onto a blue printed checkerboard.

The blue printed pattern is used to determine the intrinsic parameters of the camera and the equation of the calibration plane in the camera coordinate system. A perspective transformation can then be applied to the calibration images, which results in an image as seen from the location of the projector. The transformed red pattern can then be used to determine the intrinsic parameters and the 3D location of the projector relative to the camera (extrinsic parameters). Typical camera-projector calibration results for the extrinsic parameters of the system (with calibration board locations) are shown in Figure 16.

Once the intrinsic and extrinsic parameters are known, the object depth from images in correspondence can be calculated using the triangulation principle (Eq. 1).

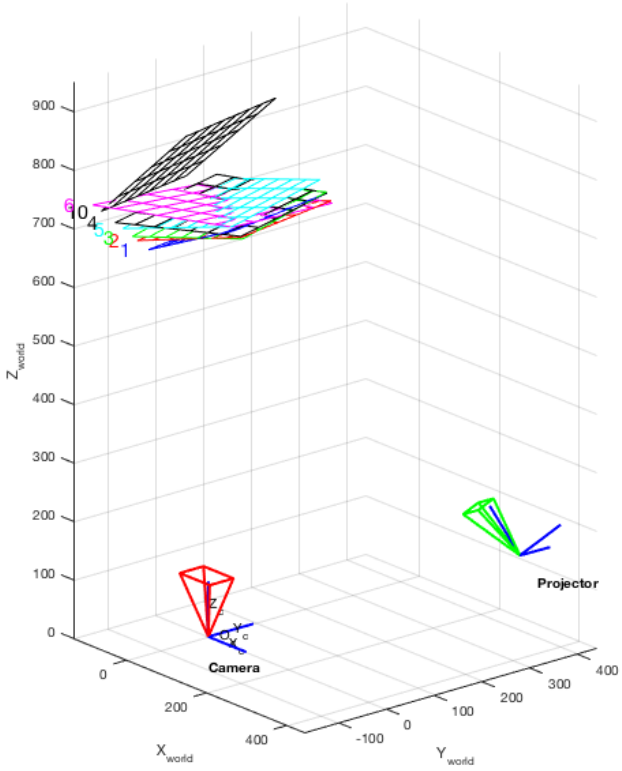


Figure 16: Calibration results - extrinsic parameters (camera-centred).

2.5 Scapular template

Charbonnier *et al.* (2014)(Table 2) presented a patient specific measurement technique for the shoulder based on the fusion of MRI data and marker based motion capture data. A patient-specific MRI model of the shoulder girdle was used to establish anatomical constraints for the entire kinetic chain of the shoulder while surface marker data across the scapula provided surface mapping data. Using the kinetic chain and surface mapping data, they calculated the location of the scapula based on a minimisation function that calculated the distance between the bone model and the measured marker positions. They reported a reduction in soft-tissue interference errors.

In the study by Mattson *et al.* (2012)(Table 2), where the aim was to validate a surface mapping method for identifying scapular kinematics, a patient-specific scapular template was used to estimate the location of the scapula by using an iterative surface-fitting algorithm. The patient-specific template was created using markers placed on the scapula on the following locations: acromion angle, approximate midpoint of scapular spine, root of the spine, approximate midpoint of medial border, and the inferior angle. The template was recorded

while the patient held his/her arm in a neutral position with the arm relaxed at the side. This template was used to inform an iterative algorithm that calculated the distance from the template to the surface map through a range of scapular positions to yield a minimum, i.e. best fit position for the scapula. This was done for six other arm positions, including abduction in the coronal plane, hand behind back, internal and external rotation as well as the hand to mouth position. They found that results for estimating the scapular position and pose were comparable to other existing methods.

It is clear from these two studies that some information about the underlying bone geometry is required to predict the scapular location when relying purely on surface mapping information. Charbonnier *et al.* used MRI bone models and Mattson *et al.* constructed a scapular template using surface markers. This study will explore the possibility of using another method for constructing a scapular template using a statistical shape model (SSM) of the scapula. A brief theoretical background of statistical shape models is presented in this chapter, while details about the technical implementation can be found in the next chapter.

2.5.1 Statistical shape model

Mutsvangwa *et al.* (2015) recently constructed a statistical shape model of the scapula. A statistical shape model is a collection of permitted variations in shape that is present in a training population. Shape data is collected for each object (e.g. the scapula) in the training population by means of a CT scan, MRI or even ultrasound in some cases. A principal component analysis is then used to parameterise the variability in shape. Once a set of parameters has been determined, these can then be adjusted to adapt and essentially fit the model to a patient specific anatomy. SSM fitting can be used to generate an accurate patient specific 3D shape representation using an extremely sparse set of 3D points (Rajamani et al., 2007).

The rationale for using a SSM is that it can potentially inform a surface-fitting algorithm that aims to determine the scapular position from a 3D topographic map of the upper back. It is possible to create a scapular template that includes a few scapular landmarks, similar to Mattson *et al.*, where the location certainty is very high, i.e. the locations have been palpated

to a certain accuracy. In addition to the points of high-certainty, the shape model can generate the rest of the scapula with a most-probable, population-specific shape, given the points of high certainty. While this does not necessarily add extra information to the scapular template, it does provide a rough shape estimation of a patient specific scapula, given the subject is represented in the SSM population. This rough estimate could potentially be used to improve the accuracy of a surface-fitting algorithm.

3 Methodology

The main objective of this study was to design and develop a tool that can be used to evaluate scapular dyskinesis and provide quantitative measurements of the scapula. In order to achieve this, the first step was to understand the clinical problem and select appropriate clinical parameters to effectively evaluate the scapula. This was done through an extensive literature review and consultations with qualified physiotherapists. The second step was to consider available technologies and select the best suited solution according to a set of design inputs.

Section I of this chapter starts with a discussion of the main clinical considerations that served as design inputs, followed by the rationale for selecting a structured light approach. This is followed by the detailed design and testing of the final solution. Section II of this chapter covers the clinical evaluation methodology and the final testing of the tool in a clinical environment.

Section I: Design Methodology

3.1 Design inputs

3.1.1 Clinical considerations

Developing a reliable clinical method of evaluating scapular dyskinesis is difficult due to several reasons. Firstly, the scapula moves in essentially two translational (assuming the thorax to be the reference) and three rotational directions. Secondly, it is challenging to accurately observe these movements beneath the overlaying muscles and soft-tissues (Matsui et al., 2006). Thirdly, the clinical assessment criteria for scapular dyskinesis are based on current visual assessment techniques where physical measurements are not the norm. It is, therefore, not clear which parameters should be measured. The main clinical considerations that influenced the system design are discussed below.

Key clinical parameters

An investigation into the diagnosis of scapular dyskinesis led to the identification of the following key clinical parameters that have proven to be valuable in diagnosing scapular dyskinesis. Figure 17 show these parameters on the scapula.

1. Scapula inferior angle prominence (Uhl et al., 2009)

   Inferior angle prominence refers to the height of the inferior angle above the thorax. This points to a scapula that no longer sits flush against the thorax but is tilted forward.

2. Scapula superior angle prominence (Uhl et al., 2009)

   Superior angle prominence refers to the height of the superior angle above the thorax. This points to a scapula that is tilted backward.

3. Scapula medial border prominence (Uhl et al., 2009)

   The entire medial border of the scapula is clearly visible. This points to a scapula that is tilted outward.

4. Lateral scapular displacement (Kibler, 1998)

   When the entire scapula, measured from the inferior angle and base of the spine to the spinal column, is displaced laterally with respect to the spine.

5. Scapula upward rotation (McClure et al., 2009)

   When the scapula shows signs of upward rotation (as described in Table 1) at rest, typically seen when comparing left to right.

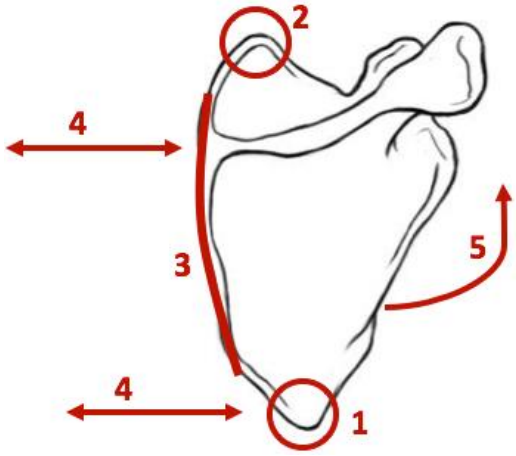


Figure 17: Key clinical parameters illustrated, image adapted from (LoveOfDrawing, 2016).

These parameters are normally present in the resting shoulder position and can usually be visually identified by a trained clinician. These parameters are commonly assessed with the arm at rest and abducted to 45° and 90° respectively (Kibler & McMullen, 2003).

Soft tissue effects

All non-invasive scapula assessment methods are affected by skin motion errors or soft tissue artefacts (STAs) because the scapula moves beneath the overlaying soft-tissue during arm movements. The specific anatomy of these soft-tissues must therefore be carefully considered when developing a measurement tool to reduce measurement errors.

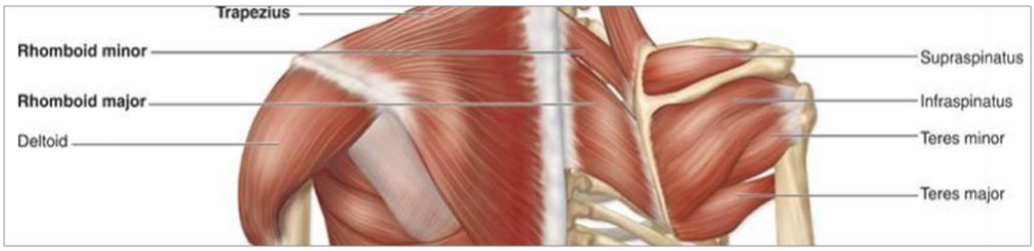


Figure 18: Part of Figure 3 with superficial (left) and deep muscles (right) shown.

Figure 18 shows the superficial (left) and deep (right) muscles that cover the scapula with the arm at rest. From scapula to the skin, the soft-tissue covering of the scapula is organised as follows. Firstly, the deep muscles (shown on the right) attach in such a way that the inferior angle, medial border and superior angle of the scapula are partially uncovered by muscle tissue. The spine of the scapula and the acromion are not covered by the deep muscles. Then the superficial muscle layer, which includes the deltoid and the trapezius both attach to the spine of the scapula. The trapezius is a very thin, flat muscle that usually covers part of the medial border and the superior angle. Depending on the individual anatomy, the inferior angle may or may not be covered by the trapezius. The spine of the scapula and the acromion is not covered with connective tissue. The final layer consists of the skin and a subcutaneous fat layer, which is usually evenly spread out across the back.

It is clear from the anatomy that the soft-tissue covering the scapula is organised in such a way that certain scapula landmarks are more exposed than others. With the arm at rest, the landmarks can be ranked from most exposed to least exposed:

1. Spine and acromion
2. Inferior angle
3. Medial border
4. Superior angle

As the scapula moves with the arm, this ranking is maintained whereas the inferior angle and medial border of the scapula become even more exposed as they move out from underneath the trapezius.

A non-invasive method for evaluating the scapula should focus on the most exposed landmarks in order to reduce the effect of STAs. In terms of accuracy, the most exposed landmarks will be the least influenced by STAs while the least exposed landmarks will be affected the most. This exposure ranking explains why the acromion marker cluster (Table 2), which is placed on the acromion for tracking the scapula, is one of the most accurate marker-based methods. It might also explain why most visual diagnostic tests for scapular dyskinesis focus on inferior angle and medial border prominences. These landmarks are among the first to become visible when the scapula position has been altered.

3.1.2 Technical considerations

Several different methods have been used to evaluate the scapula and the main methods were highlighted in the literature review. The most common methods use reflective markers placed on specific scapular landmarks (like the acromion), but these methods have several disadvantages, as discussed in section 2.3.1. In summary, the main disadvantages of marker-based methods are:

- They are prone to measurement errors due to STAs,
- they require time-intensive marker placements for every test session,
- a system-level calibration is required for every test session,
- manual marker placements hamper test repeatability, and
- they require costly multiple-camera optical motion tracking systems.

It was also clear from literature that a surface mapping approach could be used to evaluate the scapula. One approach to creating a surface map is to use multiple markers, as Mattson *et al.* and Charbonnier *et al.* did (Table 2). While these marker-based surface mapping methods attempted to reduce STA based errors, they failed to address the time, cost and repeatability components.

To address these disadvantages, a marker-less surface mapping approach was therefore proposed. It was suggested that an optical method can be used to create a surface map instead of markers. Solutions like Moiré topography, photogrammetry, 3D laser scanning or a structured light approach were considered and based on several considerations, a structured light approach was chosen for this study. The main advantages of a structured light approach over other optical methods, as well as how it improves on some of the disadvantages of marker-based methods are discussed below.

Speed[1]

In order to create an accurate 3D representation of the human body, a scanning system has to be able to rapidly capture 3D data. The body is constantly in motion due to continuous respiratory, cardiac and muscle activity. Any data collection that is longer than a few hundred milliseconds would therefore contain motion related artefacts. For example, 3D laser scanners that rely on time-of-flight technology work well on inanimate objects but create noisy data when used on the human body. This is because data collection typically takes several seconds to complete, during which the body cannot be kept completely still.

A structured light or Moiré topography scanning approach would enable rapid data collection using a single camera image typically taken at 1/80th of a second (12.5 milliseconds). This short exposure time has an added advantage in that it would enable video capture, typically done at 70 frames per second (about 143 milliseconds per frame), for dynamic 3D motion analysis.

Accuracy

Marker-based methods rely on palpation to accurately place reflective markers on bony landmarks. Palpation of the skin surface has been shown to be a reliable, non-invasive method for estimating the location of the bony landmarks of the scapula. Lewis *et al.* (2002) showed that accuracies of between 2-5mm could be achieved on the scapular landmarks. Even though

[1] This design consideration was only included after the first iteration of this project showed that a multi-shot structured light approach was insufficient. Refer to section 3.3.2. In addition to this finding, preliminary testing using a commercial 3D scanner also confirmed this requirement.

‘pin-point’ accuracy is not achievable with palpation, it does provide a non-invasive method for estimating the position of the scapula. A non-invasive method for evaluating the scapula using an optical approach should aim to be at least as accurate as the palpation method in order to provide comparable results. The accuracy of marker-based methods is not only influenced by STAs but also the accuracy of the initial marker placement by palpation.

Cost and availability

In addition to the data capturing capabilities, a structured light system can be designed, built and implemented using only a computer, a digital camera and a data projector. These are relatively low-cost items that are commonly available in most academic and clinical research laboratories. A low-cost solution would allow easier access to 3D data capture technology, especially in academic laboratories or clinical settings where budgets are limited.

Cost is primarily driven by the hardware requirements and when a structured light implementation is compared to some other methods the cost of such a system is dramatically reduced. Marker based methods, for example, require expensive multi-camera systems. Laser scanning systems require only a laser scanner connected to a computer, but the cost of the scanner is directly related to the scan speed, which is important when scanning the human body. A commercial structured light scanner that has been proven to work on the human body can cost in order of R135 000 (Artec 3D, 2017). A mid-range 3D laser scanner (with a scan timeframe of about 7 seconds) costs about R110 000 (iReviews, 2014a) and a high-end laser scanner with a faster scan-rate can cost in the order of R350 000 (iReviews, 2014b). An affordable Optitrack 8-camera marker-based system costs about R130 000 while a high-end system costs about R750 000 (Optitrack, 2017). A comparable Vicon 8-camera system can cost three to four times as much (Carse, Meadows, Bowers, & Rowe, 2013).

Photogrammetry is another low-cost option that relies on more than one 2D image, simultaneously taken from different positions, to reconstruct 3D objects using triangulation. This method often involves two or more cameras that trigger at the same time. The disadvantage with this method is that it relies on the visual features (like sharp corners or dark lines) of the scanned object to establish correspondence and calculate the depth map, this becomes difficult when the object has no easily identifiable features, for example a smooth

skin surface. Some scanners make use of projected light patterns and Moiré topography to assist with feature detection on smooth surfaces (Artec 3D, 2017). The system then essentially becomes a hybrid of several methods and the cost of the final solution increases.

Moiré topography is also a low-cost solution that is very similar to the structured light approach where shadows (or projected light) is used to create artificial texture on a scanned object. However, as discussed before, these methods are less commonly found in literature and implementations are not as well documented as structured light methods.

Setup time and calibration

Marker-based methods require time-consuming manual marker placements where individual reflective markers have to be placed on palpated bony landmarks. Once the markers have been placed on a subject the multi-camera system needs to be calibrated before testing can begin. If the markers are removed for any reason, the whole process must be repeated. Optical methods, like structured light imaging, also require a system calibration, but this calibration is independent of the test subject. Once the system has been calibrated, multiple tests on multiple subjects can be done. Because optical methods rely on photos rather than markers, re-calibration is only required when the optical centres of the equipment change, for example when the equipment is moved. With optical methods, less time is spent on system calibration than with marker-based methods, especially when a large test population is required.

Resolution

The resolution of a surface map created using a marker-based method is much lower when compared to a structured light or Moiré topography surface map. A marker-based surface map has one data point for every marker, whereas an optical method produces one data point for every pixel in correspondence. Charbonnier *et al.* (2014) used 56 markers to create their surface map, while Matson *et al.* (2012) used 300 markers placed on Kinesio tape. These data points need to be interpolated in order to create a surface, and even when using 300 points the resolution is very low considering that the area across the scapula can be 400-600cm^2. By

using a structured light approach, a surface map with tens of thousands of data points can easily be created.

Repeatability

In addition to the time and resolution improvements that an optical surface mapping method offers, it also produces highly repeatable raw data. Two optical camera images of the same subject, taken on different days, should be able to reliably produce the same surface map, if the system was designed to accurately generate a surface map using only a camera image. The only difference in the two surface maps should be the subject-specific changes. The differences in the surface map could then potentially be used to compare pre- and post-treatment states. On the other hand, marker-based methods rely on manual session-to-session marker placements that subject the raw data to marker placement errors that makes subject-specific repeatability difficult.

3.2 Design specifications

In summary of the clinical and technical design inputs, the design specifications are shown in Table 3. These specifications capture the most important design features that were used to evaluate the success of the final design.

Table 3: Design specifications.

Req. ID	Requirement	Short description	Specification
01	Cost	Hardware costs should be affordable. The estimate is based on 10% of a commercial structured light scanner of R135 000.	< R13 500
02	Setup time	Marker based systems can take 1-2 hours to set up. A comparable timeframe would be ideal.	2 hours or less
03	Calibration	The ease of use and additional equipment requirements for calibrating the system.	Easy to do with minimal training
04	Speed	The scan rate of the system must be high in order to avoid movement artefacts. Video analysis is possible at 70 frames per seconds.	1/70 seconds or faster
05	Scan area	The entire area across all the scapular landmarks should be included in a single scan. A typical scapula area is 20cm x 30cm = 600cm^2.	600cm^2 or more
06	Resolution	The area across the scapula is typically 400-600cm^2. Ten data points per square cm is chosen and would provide a 20x improvement on the best marker-based method.	4000-6000 data points across the scapula
07	Accuracy	The ability to make precise measurements of a 3D object must be at least as accurate as measurement by palpation.	2-5mm or better
08	Repeatability	Deviation in multi-session measurements should be minimal. 20% or less deviation is chosen as this allows for 1mm variability on a 5mm accurate measurement.	20% deviation

3.3 Detailed design: structured light scanner

Our structured light scanner consists of an off-the-shelf camera and projector, a custom-made calibration board and software components. The hardware requirements are relatively simple and can be assembled with some prior knowledge of optical and structured light systems. The software components were the most difficult to develop and a large part of the system design effort was therefore devoted to software design.

The main software components were designed according to the principles outlined in an article by Zhang *et al*. (2002) “Rapid Shape Acquisition Using Colour Structured Light and Multi-Pass Dynamic Programming”. An overview of the structured light system architecture is shown in Figure 19 and the main functions are discussed in the rest of this section. The

corresponding section numbers are shown and the Matlab function names are given in square brackets for easy review of the Matlab code attached in Appendix I: Matlab code.

A 2009 MacBook Pro with a 2.53 GHz processor, solid state hard drive and 8GB memory was used for this project. The software components were developed using Matlab (64-bit version R2016a) with the following toolboxes installed: Computer Vision System Toolbox (v7.1), Image Processing Toolbox (v9.4), Signal Processing Toolbox (v7.2) and Curve Fitting Toolbox (v3.5.3). In addition to these built-in toolboxes, a few other third-party functions were used and referenced in the Matlab code.

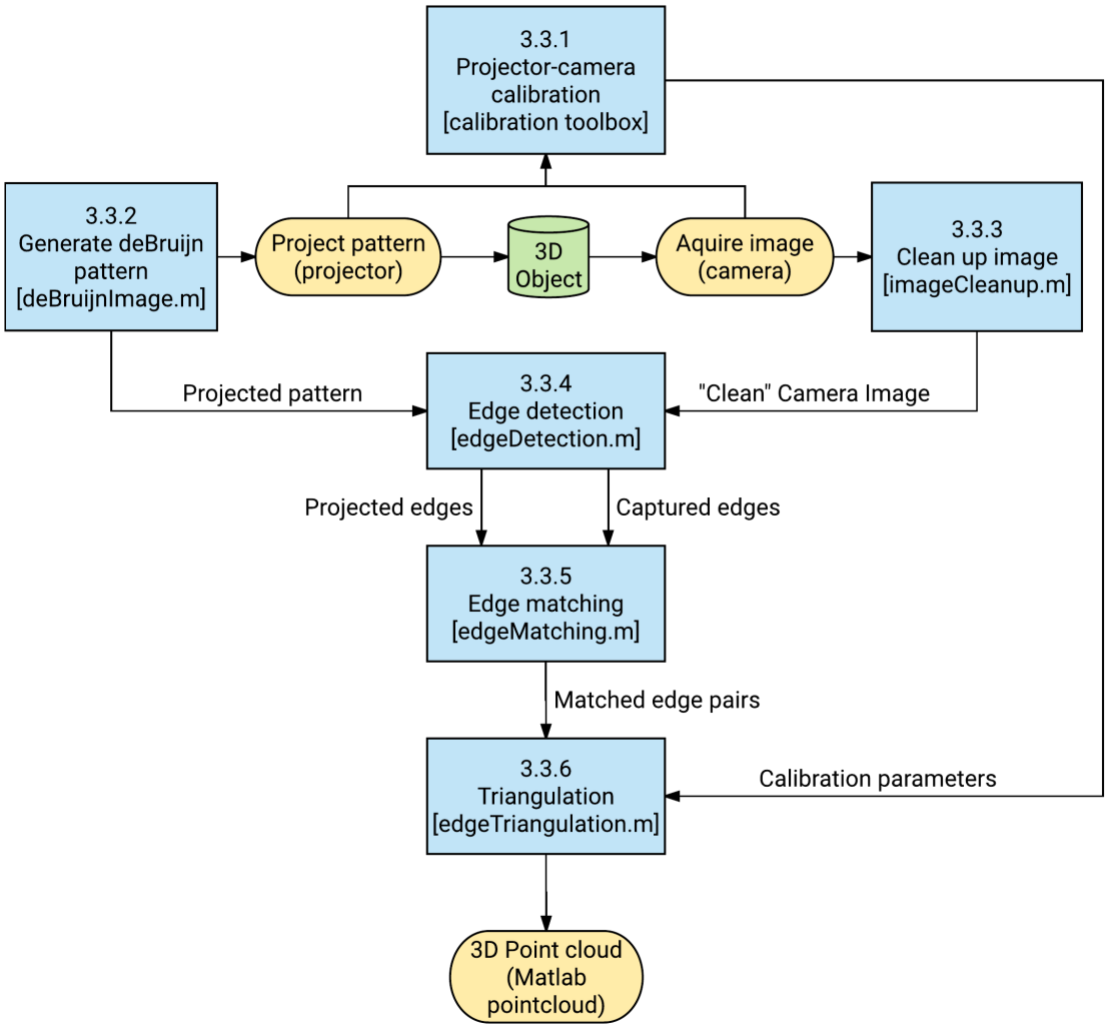


Figure 19: Overview of the system architecture.

3.3.1 Projector-camera setup and calibration

A Canon EOS 600D SLR camera with an EF-S18-55mm f/3.5-5.6 lens, set to a resolution of 3456×2304 pixels, was used in combination with a NEC NP100 projector, set to a resolution of 800x600 pixels (projector maximum). The camera was placed on a high-quality, stable tripod with adjustable angle and height, the projector was placed on a stable surface. A typical setup configuration is shown in Figure 20.

Once the camera and projector was set up, a calibration board was used to determine the extrinsic parameters of the system. The mathematics of the calibration process assumes that the calibration pattern is in a perfect plane, therefore the calibration board needed to present the checkerboard pattern on a surface that is as flat as possible. Most examples of calibration boards found in literature consisted of a printed paper glued to a wooden or plastic surface. This construction method introduces errors into the calibration because most low-cost wooden or plastic objects do not have completely flat surfaces, there is always some surface deformation or curvature present. These deformations can be removed through a high-precision machining process, but this is costly.

Figure 20: Camera-projector setup.

Ordinary float glass, similar to household windows, was used for this project in order to improve the calibration board design and create a board that is extremely flat. Float glass is manufactured using a process where molten glass is floated on a bath of molten tin in a chemically controlled atmosphere (PFG Building Glass, 2017). This process creates glass with a faultless finish and a near-perfect flatness. The checker-board pattern was printed from a high-resolution image on A3 paper and pressed flat between two panes of float glass, held together tightly using strong adhesive tape. The calibration board is shown in Figure 21. The flatness of the glass panes pressed together ensures that the printed pattern was held in a near-perfect flat configuration.

Figure 21: Calibration board.

Once the hardware components had been assembled, several other calibration parameters needed to be considered, like:

- the amount of calibration observations to use,
- the angular range of the calibration board during calibration, and
- the optimal view-angle setup for the camera and projector.

The study, "Precision and accuracy parameters in structured light 3D scanning" by Eiríksson *et al.* (2016) proved valuable in selecting good calibration parameters. The accuracy of the calibration is partly based on the number of calibration images taken, therefore it is essential to take several pictures of the checkerboard in different positions. Eiríksson *et al.* found that 11 calibration images provided optimal results for a structured light system whereas using more images yielded negligible improvements. They showed that an angular range of ±5° up to ±40° for the calibration board position produced good results.

For this project, calibrations therefore typically included 11-15 observation images with the calibration board moved through a minimum of ±20°. The optimum view angle between the camera and the projector was found to be around 20-30°. The projector was placed at an angle with the camera in both the vertical and the horizontal planes.

3.3.2 Encoding strategy

Once the camera and projector had both been calibrated, the next step involved choosing the appropriate encoding pattern to project onto the scanned object. This process is often done iteratively to optimise the project-scan-reconstruct process. This section outlines the main considerations and design process for developing the best encoding strategy for our application, i.e. scanning the human body.

Binary coded multi-shot pattern

The easiest encoding strategy to implement programmatically, and therefore the first iteration for this project, was a multi-shot sequential binary-coded pattern (Figure 22). Black and white stripes were used to form a sequence of projection patterns that cover points on the object with a unique binary code. Pixel correspondence could be extracted from the coded pattern and used for triangulation. It provided a very reliable method for establishing correspondence since only dark or light values exist at all the pixels. The drawback, however, was that many projection patterns were required to achieve a high spatial resolution and the scanned object had to remain completely static for the duration of the image acquisition process.

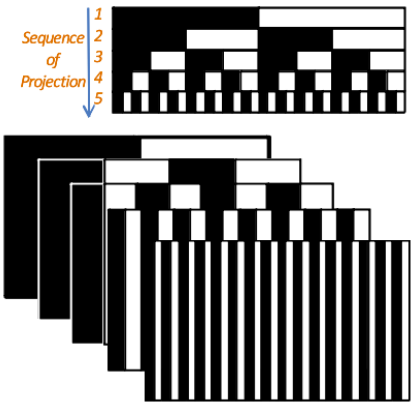
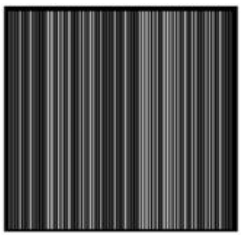
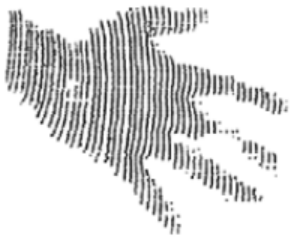
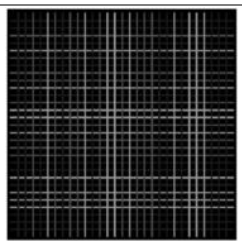
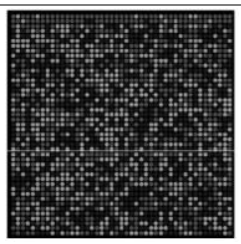


Figure 22: Sequential binary-coded pattern projections (Geng, 2011).

A binary encoding strategy using 21 vertical and 21 horizontal projections (42 in total) was used for the first implementation of this project. Initial testing revealed that the image acquisition process took approximately 30-60 seconds and produced a very noisy reconstruction due to the continuous movements of the human body during scanning. The first design iteration proved that a single-shot encoding strategy would be needed to scan the shoulder.

Salvi *et al.* (2004) presented a comprehensive survey of coded structured light techniques in "Pattern codification strategies in structured light systems". They implemented and tested seven of the most prominent encoding strategies and reported on the quantitative and qualitative results in detail. Table 4 summarises the results for the three best single-shot strategies.

Table 4: Single-shot encoding strategies comparison results (Salvi et al., 2004).

Encoding strategy	**Pattern example**	**Reconstruc-tion error Stdev (μm)**	**Number of 3D points**	**Reconstruction of a human hand**
deBruijn pattern with 64 colour bands		13.1	13899	
Colour Grid pattern with 29x29 slits		72.3	372	
Colour dot pattern with 45x45 elements		23.6	926	

A deBruijn pattern encoding strategy was chosen for this project primarily based on the superior accuracy and resolution of the method compared to other single-shot strategies. The colour and surface texture of the scanned object are also an important consideration for

choosing the right encoding strategy. A deBruijn colour pattern is well suited for scanning a smooth, mono-colour surface, like human skin. In addition to these advantages, implementation of deBruijn pattern strategies are well documented in literature (Van Kessel, 2013)(L. Zhang et al., 2002)(Li, 2004).

DeBruijn pattern generation

In order to create correspondence using colour bands, the edges of these colour bands had to be uniquely identifiable. Using the RGB colour channels, we could effectively create eight different colours if each colour was represented by either 1 or 0. For example, red (1,0,0), blue (0,0,1), yellow (1,1,0), magenta (1,0,1), etc. Each colour channel was limited to its extreme values (1 or 0) to make the colour distinctions clearer.

Colour bands of equal width were projected and the transition from one colour to the next was recorded. For example, when blue (0,0,1) transitioned to red (1,0,0) the change could be noted as: (0,0,1) → (1,0,0) = [+1,0,-1]. The red channel changed from 0 to 1 and the change was indicated by a [+1] on the right, the green channel remained at 0 with no change shown as [0] and the blue channel changed from 1 to 0 and the change was indicated by a [-1] on the right. Using these transitions in three channels created $3^3 = 27$ unique transition values. This effectively created 26 unique encoding possibilities as the [0,0,0] value refers to a non-transition. Using only 26 edges, however, would not create a good resolution over a large object. One way of solving this was to repeat the colour pattern many times, but this complicated the process of finding correspondence in the camera image because multiple matches could be found. To minimise such ambiguity, a windowed uniqueness approach was proposed by Zhang *et al*. (2002). The approach proposed a sequence of colour stripes that contained sub-sequences (or windows) of edge transitions that were unique within the entire sequence. A deBruijn sequence is well suited for this.

A deBruijn sequence is a string of characters that is made up of k unique letters or numbers, arranged in such a way that every window (or substring) of length n will be unique in the entire string. The length of the sequence (L) depends on the number of unique letters or numbers used (k) and the desired window size (n) and can be expressed as $L = k^n$.

In order to create a deBruijn sequence, the first step was to select the window size required to detect features and establish correspondence. Secondly, the number of unique letters, or colours in this case, needed to be selected. DeBruijn sequences are not straightforward to derive and therefore a modified online Matlab sequence generator was used (Brimijoin & Geest, 2011). Once the sequence was generated, the colours could be arranged in such a way that the edge transitions in each window was unique throughout the entire sequence. A classic deBruijn sequence allows the same character to be placed successively, so as to ensure that the same colour is not repeated, some extra operations are required.

Zhang *et al.* generated a sequence with an XOR logical operator to ensure that all edge transitions produced a change in at least one colour channel. Their final sequence used $k = 5$ different colours with a window size of $n = 3$, producing a sequence with 125 stripes.

For this project, 6 colours were used. Black (0,0,0) and white (1,1,1) were left out because initial testing revealed that it could be confused for the colour of the skin in certain ambient lighting conditions. To improve on Zhang's method, colour selections were done in such a way as to produce edge transitions that change in at least two of the three colour channels vs. only one as in Zhang's case. This improved the robustness of the edge detection algorithm later in testing. Doing this effectively reduced the number of unique colours that could be used in a window to $k = 3$. Therefore, a larger window size was needed to produce enough stripes. A colour stripe width of 7 pixels was chosen based on the work by Pages *et al.* (Pagès, Salvi, Collewet, & Forest, 2005), and to generate enough stripes for an 800x600 projected image, a window size of at least $n = 5$ was needed. Window size is typically chosen based on the expected fine features on the scanned object, because human skin represents a smooth surface, a larger window size was justifiable. A section of the projected deBruijn pattern created for this project is shown in Figure 23.

Figure 23: Section of the projected pattern used for this project.

3.3.3 Image clean-up

CCD cross-talk correction

The encoding pattern from the previous section was projected onto the object and a camera used to capture the coloured light reflections. The Bayer filter over the CCD separates the light into red, green and blue sections. Due to the nature of these Bayer filters the light is however, not separated perfectly and each colour channel is influenced by light reflections from all the other channels. This phenomenon is known as CCD colour cross-talk.

The colour detection was further complicated by the fact that each person's skin reflects light in a unique spectrum. Some skin types are more absorbent and have darker appearance while other skin types reflect more light and appear lighter. In addition to this, every skin type has a unique reflection pattern across the colour spectrum and will reflect colours differently. In order to correct for this, a two-fold approach was followed.

Firstly, to correct for cross-talk, Caspi *et al.* (1998) proposed a model that relates the camera image colours S to the projected colours P:

$$\begin{matrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \\ S \end{matrix} = \begin{matrix} \begin{bmatrix} a_{RR} & a_{RG} & a_{RB} \\ a_{GR} & a_{GG} & a_{GB} \\ a_{BR} & a_{BG} & a_{BB} \end{bmatrix} \\ A \end{matrix} \begin{matrix} \begin{bmatrix} k_R & 0 & 0 \\ 0 & k_G & 0 \\ 0 & 0 & k_B \end{bmatrix} \\ K \end{matrix} \begin{matrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} \\ P \end{matrix} + \begin{matrix} \begin{pmatrix} R_O \\ G_O \\ B_O \end{pmatrix} \\ O \end{matrix} \tag{4}$$

where A is the camera-projector colour cross-talk matrix, K is known as the albedo matrix for a point on the object and O represents the ambient light present at that same point. The cross-talk matrix can be determined by projecting a red, blue and green scene onto a white surface in zero ambient light conditions and capturing three camera images. The average of the RGB values is used to construct A. Matrix K can be determined by projecting uniform white illumination onto the object. Colour-corrected camera image values can be obtained by multiplying each camera colour channel with the inverse of the cross-talk matrix, when the ambient light is assumed to be negligible:

$$\acute{S} = A^{-1}S = \begin{pmatrix} k_R r \\ k_G g \\ k_B b \end{pmatrix} + A^{-1}O \tag{5}$$

The above method requires several reference images for each object scene. In order to improve on this, several experiments with different object surfaces were done to develop a better colour correction strategy. Our newly developed method requires no reference images and provides a simplified, yet powerful colour correction that automatically takes the ambient lighting conditions into account.

Colour correction is done by taking colour samples of red, green and blue sections in the captured image of the object under illumination. Each sample is representative of the respective colour and compared to the known projected values. For example, the red sample might look like this: (R,G,B) = (0.745, 0.115, 0.077), and when compared to the projected image, which we know is (R,G,B) = (1, 0, 0), it is clear that the green and blue channels contained noise due to cross-talk, ambient lighting and object reflectance. Note that this approach only works because the object under illumination, i.e. the human skin, is assumed to have relative uniform reflectance. The three samples are used to construct a simplified "cross-talk" matrix by taking the channel averages with the known projected values:

$$A_{simplified} = \begin{bmatrix} R_1^{avg} & R_2^{avg} & R_3^{avg} \\ G_1^{avg} & G_2^{avg} & G_3^{avg} \\ B_1^{avg} & B_2^{avg} & B_3^{avg} \end{bmatrix} \tag{6}$$

Finally, the colour corrected camera values are determined by multiplying the camera values with the inverse of $A_{simplified}$:

$$\acute{S} = {A_{simplified}}^{-1} S \tag{7}$$

An example is shown in Figure 24 where colour correction was applied to part of the camera image. Note the improved colour intensity and saturation which made the colours easier to identify.

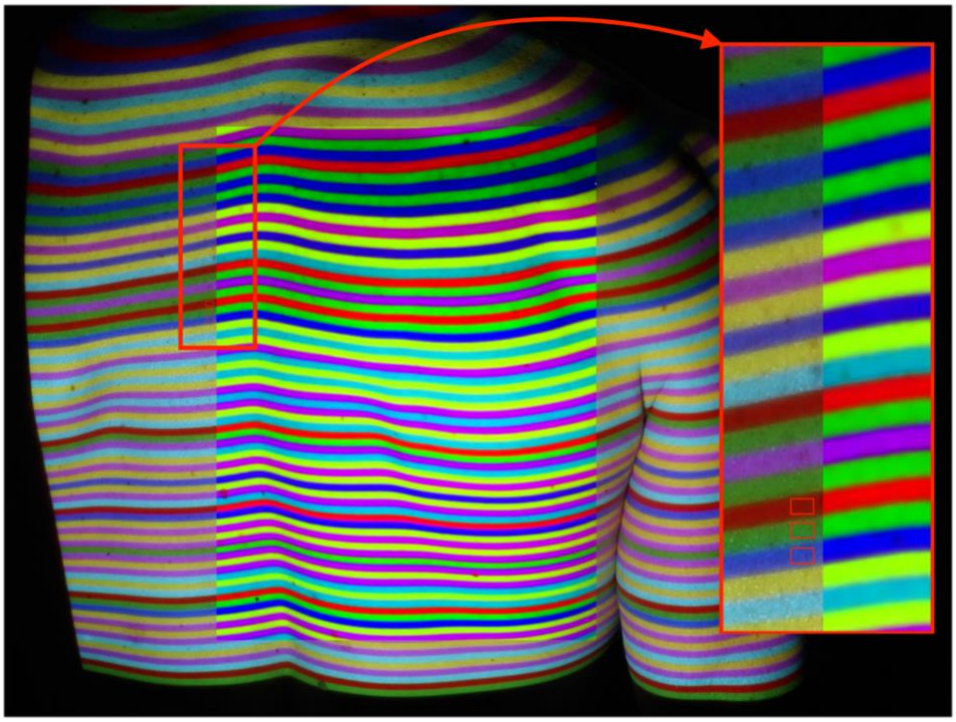


Figure 24: Camera image with colour correction applied to part of the image. Typical colour selection samples (three red blocks) are shown bottom right.

The samples used to construct $A_{simplified}$ is represented by the three small red blocks shown in the bottom of the enlarged section on the right. These samples could be selected anywhere on the image. For example, selecting darker samples will typically produce brighter colours while lighter samples tend to darken the final image. This simplified method of colour correction is far from perfect, but with some experience a user can make good sample selections that lead to excellent results in most cases. The main advantages of this simplified method are that no extra reference images are required, and the user has some added flexibility to deal with variations in ambient light and object reflectance.

Noise removal

The final step before the colour-corrected image could be used for edge detection was to apply a smoothing filter. Because resolution of the camera is much higher than that of the projector, the individual pixels of the projected image are often visible. To remove these pixel borders, as well as any other image noise, a noise filter is used. Such a filter must be carefully selected to ensure that it removes the noise without changing the locations of the colour transition edges. The edge locations are used to establish correspondence and their locations must be recorded as precisely as possible to ensure accurate 3D reconstructions.

After testing several different filters, a non-local means filter (NLMF) produced the most desirable results. A Matlab implementation developed by Kroon was used in its default configuration (Kroon, 2010). Figure 25 shows the filter results. Note that the colour bands are smooth and without noise while edge locations are not altered in any way. The bottom colours appear brighter because the colour correction and NLMF are applied together. The filter effect can therefore also be seen in Figure 24.

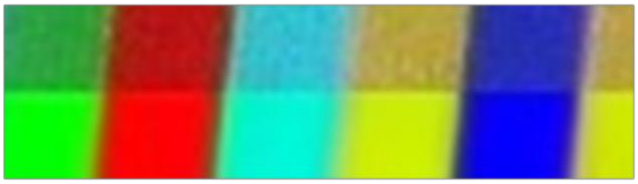


Figure 25: Non-local means noise filter, before (top) and after (bottom).

3.3.4 Edge detection

The main challenge with any structured light implementation is to establish correspondence between the projected image and the camera image. Figure 26 shows the projected pattern (bottom) and the camera image (top).

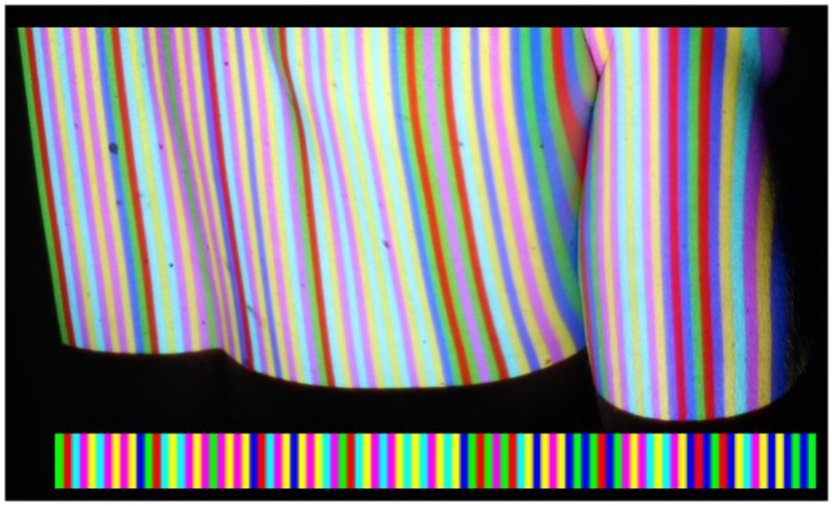


Figure 26: Camera image with projected pattern shown.

In our case the colour edges are used to establish correspondence and while it is relatively easy to visually match the colour transitions in Figure 26, it is not a trivial task to match the edges mathematically. Firstly, all edges need to be detected in both the captured and projected images, then every edge in the camera image needs to be matched up with the same edge in the projected image.

To detect the colour edges, a gradient function is computed for each row of the image. This gradient function consists of the sum of the squares of all three colour channels. The local maxima of the gradient function are used to find the edges. Figure 27 shows the result of the edge detection for a section from a camera image. It is important that the detected edges accurately represent the colour transitions and iterative experiment-based improvements to our peak detection algorithm enabled accuracies of one to two pixels for most images. The excellent edge detection results can be clearly seen in Figure 27.

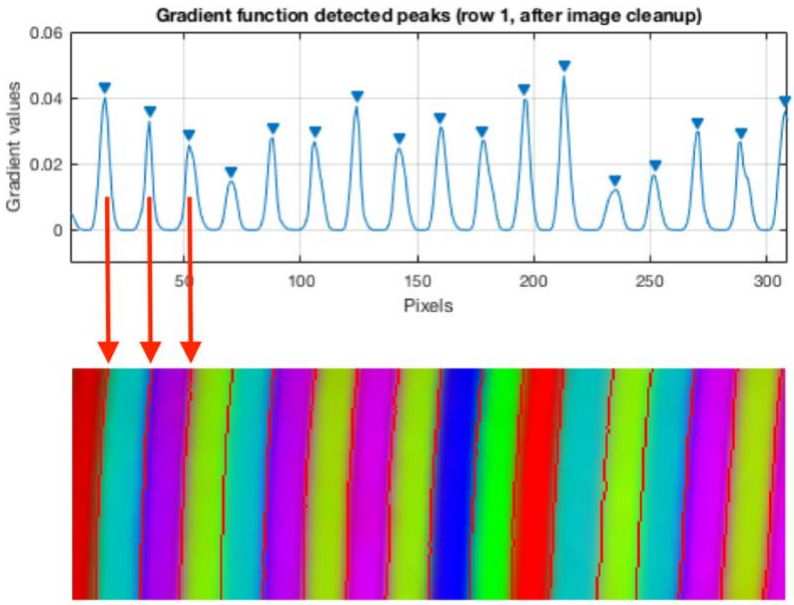


Figure 27: Edge detection results for the first pixel row, peak detection (top) and edge locations (bottom) shown in red.

3.3.5 Edge matching

In the ideal case, all the projected edges are all visible on the camera image and a simple edge matching algorithm can be used to link both edge sequences. Unfortunately, the complete projected pattern is rarely captured and shadows on the object may remove some edges mid-sequence. To solve this, a deBruijn pattern with windowed uniqueness is employed. The location in the colour pattern can be determined using only five consecutive colour transitions, i.e. one unique window. In addition to an incomplete pattern, edge matching is further complicated by falsely detected edges on the camera image. False edges can be detected when the CCD observes a colour change that is not produced by the projected

pattern. A colour change can be caused by surface anomalies like changes in texture or surface reflectance. It is therefore important to be able to distinguish true edges from false edges.

To correctly match the detected edges and exclude these false edges, Zhang *et al.* (2002) proposed a scoring function $score(p, c)$ that evaluates a match between a projected colour transition p and an observed edge c in the camera image. The scoring function scans through the camera and projector images row by row and compares the gradient (change in intensity) of all three colour channels at each edge transition. A projected edge is matched to the camera edge only when the changes in all three colour channels for both match. The projector image edges provide the ground truth to which the camera edges are compared. The scoring function is defined as follows:

$$score = \min\{consistency(p^{[r,g,b]}, c^{[r,g,b]})\} \tag{8}$$

where $consistency(q^c, e^c) \in [-1,1]$ is used to measure the agreement between the colour channels of q and e. For example, if $p^{[r]} = +1$ (representing a positive change in the red channel), $consistency(+1, c^{[r]})$ should be 1 if $c^{[r]}$ (the change in the red channel for the camera image) is large enough, 0 if $|c^{[r]}|$ is too small and -1 if $c^{[r]}$ is negative. For the cases where $p^{[r,g,b]} = 0\ or\ -1, consistency$ is defined to also produce a +1 when the change in $c^{[r,g,b]}$ matches the change $p^{[r,g,b]}$ and a low value for a mismatch. The function is formally defined as:

$$consistency(+1, c^{[r,g,b]}) = CLAMP\left(\frac{c^{[r,g,b]} - \alpha}{\beta - \alpha}\right) \tag{9a}$$

$$consistency(0, c^{[r,g,b]}) = CLAMP\left(1 - \frac{|c^{[r,g,b]}| - \alpha}{\beta - \alpha}\right) \tag{9b}$$

$$consistency(-1, c^{[r,g,b]}) = CLAMP\left(\frac{-c^{[r,g,b]} - \alpha}{\beta - \alpha}\right) \tag{9c}$$

where

$$CLAMP(x) = \begin{cases} -1 & x < -1; \\ x \quad if & -1 < x \leq 1; \\ 1 & x > 1; \end{cases}$$

and $0 \leq \alpha < \beta \leq 1$ are user defined thresholds to set the range wherein an edge can be matched to allow for some uncertainty in the edge measurements. CLAMP is a function that allows the consistency values to be clamped between -1 and 1. The $score$ function essentially

produces a +1 for a perfect match in all three colour channels and a -1 for a mismatch in any one of the three channels.

This score is used to compare all the edges in each row of the camera image to the projected pattern and establish edge correspondence by using a path optimisation algorithm. In order to calculate the edge correspondence, Zhang *et al.* (2002) proposed a multi-pass path optimisation algorithm. The algorithm is complicated to implement and suffers from long running time due to the multi-pass approach. Van Kessel (2013) reported a run-time of 10-15 minutes to calculate correspondence for a single scan using Zhang's implementation. For this project, a simplified algorithm was used to calculate the optimal path to establish correspondence. A simpler algorithm was possible because the scanner is designed to scan the human body.

The human body, and especially the skin across the scapula, generally presents a smooth surface and it can therefore be expected that the scanner should not encounter any sharp edges or large occlusions. Furthermore, human skin is usually quite uniform in colour and even when blemishes are present, they are usually only a few shades darker than the average skin colour. As a result of these factors, the scanned object (human skin) presents shape and colour variations that are very predictable. This predictability allowed the dynamic programming algorithm to be simplified from a multi-pass to a single-pass solution.

Firstly, edge-detection was simplified and more robust due to improved colour correction and noise removal techniques. This resulted in overall greater edge predictability. Secondly, by taking the most likely object shape and colour variations into account and utilising the deBruijn pattern to initiate a sequential pattern match, the detected edges could be pre-arranged into a 'most-likely' fit. This allows our edge matching algorithm to match the edges with a single pass instead of multiple passes. This simplified implementation allows edge detection and edge matching to be computed in approximately 30-60 seconds. The result of the edge detection and matching phase is a set of projector-camera edge pairs that are stored in a file and used to calculate a 3D point cloud through triangulation.

3.3.6 Triangulation and point cloud

Once the intrinsic and extrinsic parameters from the projector-camera calibration is known and the correspondence between the camera and projector images has been established, a 3D point cloud can be computed using triangulation. This point cloud consists of all the points in correspondence and will therefore only include the colour stripe edge locations.

The straight colour edge lines on the projected image represent vertical planes projecting from the optical centre of the projector, through the pixel lines into 3D space. These lines can therefore be used to calculate 3D plane-equations for every colour edge using a least-squares fitting algorithm. For the camera image, a ray-equation is calculated for every pixel in correspondence. These ray-equations represent straight lines projected from the optical centre of the camera through the pixels into 3D space. The point where a camera pixel ray intersects a corresponding projector plane represents a 3D point on the scanned object (Figure 28). The Matlab functions used for plane fitting, ray-equation and ray-plane intersection calculations were adapted from functions created by Lanman (2009).

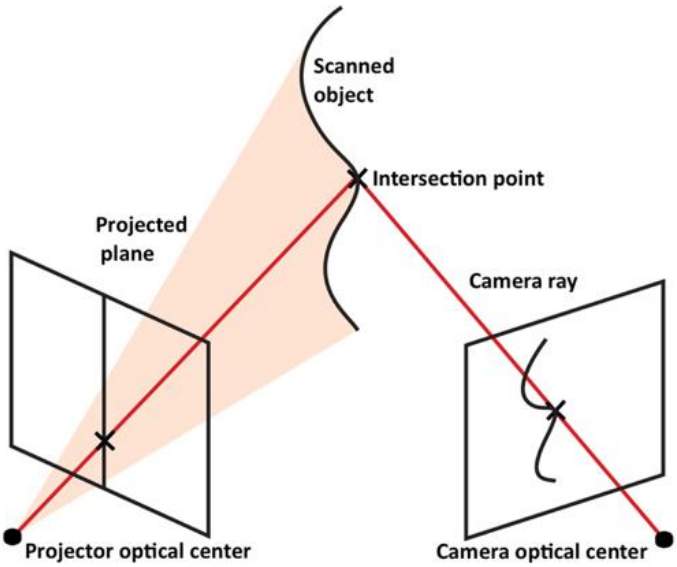


Figure 28: Ray-plane intersection, adapted from (Lanman, 2009a).

Finally, to produce a complete 3D point cloud of the scanned object, all the ray-plane intersection coordinates were calculated and stored in a point cloud data structure. This data structure allows for easy 3D graphical representation of the data and it simplifies other post-processing activities like de-noising, meshing and surface-fitting. Figure 29 provides an

example of a camera image with colour correction and noise removal applied to a section before that section was reconstructed into a point cloud.

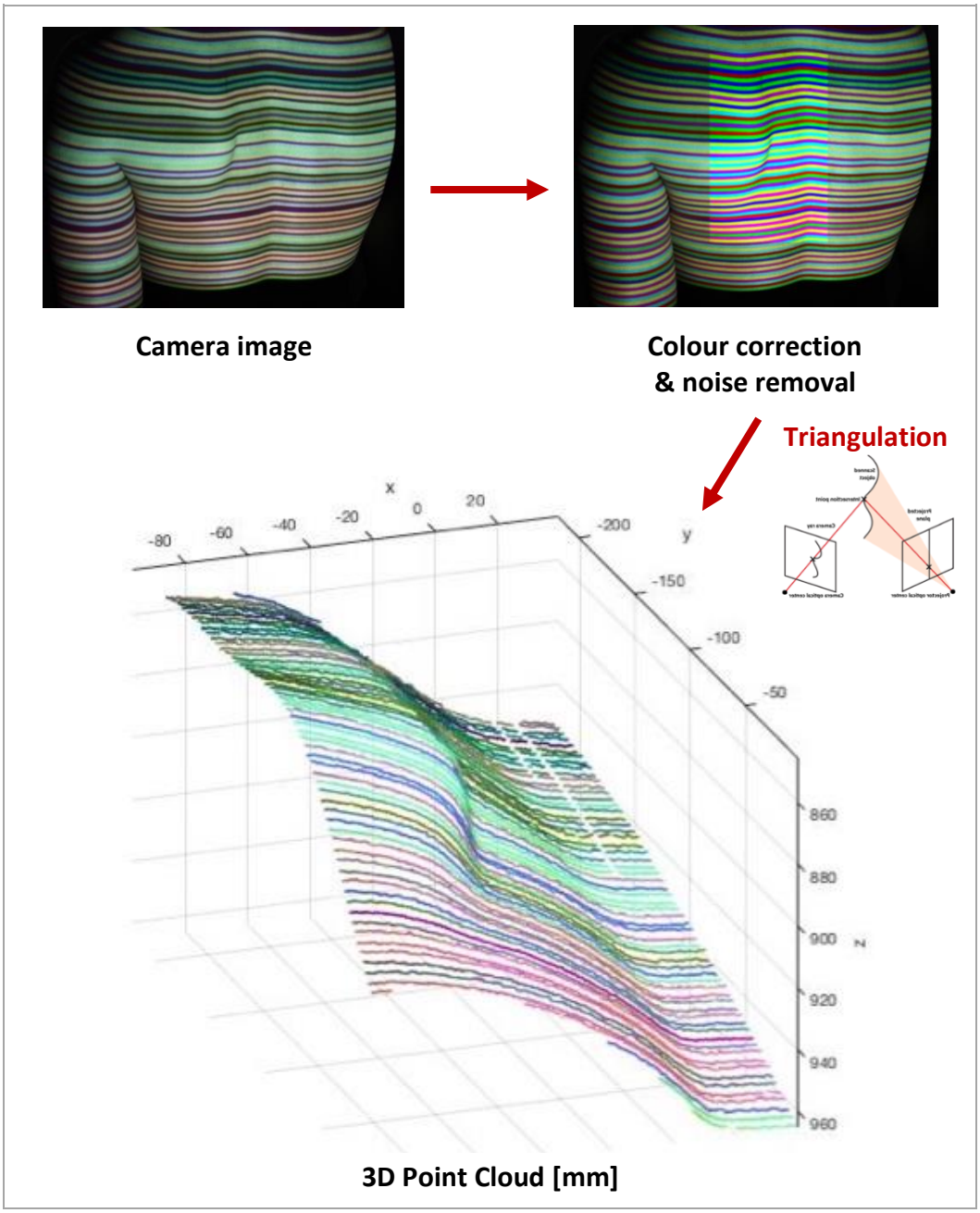


Figure 29: Process illustration - from camera image to point cloud.

3.4 Laboratory testing of developed structured light system

A series of laboratory tests were performed to evaluate the structured light system described in the previous section. The goal was to evaluate the system against the design specifications (from section 3.2) and test the system using set-ups designed to simulate the human body. Laboratory tests are done prior to any clinical evaluations to refine the final system before testing on humans.

3.4.1 Cost, setup time and calibration

The final structured light scanner, which included the camera, tripod, projector and calibration board cost less than R10 000 to build. Future implementation will be achievable with a smaller budget using a slightly less expensive digital camera and a basic projector. The resolution of the final reconstruction is limited to the projector resolution and it is therefore recommended that the digital camera resolution must be at least double that of the projector to ensure good results. The cost of the laptop computer was ignored based on the assumption that most institutions have access to a portable computer and because other methods like laser and marker-based systems also require a computer (not included in the purchase price) to operate the software.

The setup and calibration of the scanner can be done in less than two hours. The calibration process is simple and with some training anyone will be able to do it. An in-depth technical knowledge of the system is not required to use it effectively.

3.4.2 Speed, scan area and resolution

The ideal image acquisition rate was found with the camera shutter speed set to 1/80 seconds. This is faster than the 1/70 capture speeds required for video analysis. The shutter speed is influenced by the amount of ambient lighting that is present and the photographic sensitivity of the CCD (ISO). The scanner was tested in various different light conditions, from a completely dark room to a well-lit office. A light meter (Goldilux GAL-2H) was used to measure the illuminance (light level) of the test environment in lux (1 lumen/m^2). For reference, office lighting measures around 400-500 lux, an office building hallway at around 80-100 lux and a room with drawn curtains at around 5-10 lux. The ideal image acquisition environment was

found to be at around 0-100 lux before the ambient lighting starts to influence the results. With the shutter speed fixed at 1/80, the amount of light that the CCD is exposed to can be managed by adjusting the ISO number and the aperture stop size.

With the scanner set up at about one meter from the object, a scan area of 35cm x 45cm = 1575cm^2 was feasible with a full frame resolution of 68000 data points. A projector image of 800x600 pixels is divided into colour stripes that are 7 pixels wide, this gives $600 \div 7 \approx 85$ stripes with a length of 800 pixels, therefore $85 \times 800 = 68000$ pixels in correspondence or data points. This calculation holds true for a landscape (800x600) or portrait (600x800) scanning orientation. The human scapula will however, only fill about half of the frame area and the expected resolution would therefore be about $68000 \div 2 = 34000$ data points. This results in about 57 data points per cm^2 for a 600cm^2 scapula area.

3.4.3 Accuracy and repeatability

Eiríksson *et al.* (2016) confirmed the accuracy of their scanner by using two measurement strategies. The first was a flatness test where a very flat surface was scanned and a 2D plane is fitted to the reconstruction. The flatness root-mean-square (RMS) error can be calculated by comparing the reconstruction to a perfect 2D plane. The second measurement involved a 3D object with known dimensions, the object was scanned and compared to the original object. The same two strategies were used to validate the accuracy for our scanner.

For the flatness test, Eiríksson *et al.* achieved RMS errors of 100-400μm using their scanner and showed that this was comparable, and in some cases better than a commercial metrology grade structured light scanner. For this project, it was decided to use the flatness test, not only as a method to confirm the accuracy, but also as a pre-scan check to ensure that every calibration was successful. The flatness test is easy to do and can be done with the scanner in the standard setup position. The back of the calibration board was used as a flat surface because it provides a very flat, white surface (as discussed section 3.3.1). Our scanner typically achieved flatness RMS errors of about 200-400μm with repeated calibrations in different locations and set-up positions. The RMS errors were comparable to what Eiríksson *et al*. found and the results confirm the ability of the calibration process to produce highly accurate results

consistently. When a repeatability of 20% is used, as per the design specification, a <400μm repeatability would be sufficient for 2mm system accuracy, which is the lower end of the accuracy specification for our system and thus satisfies the requirement.

As a general rule, a flatness RMS error above 400μm was used as an indication that a calibration was unsuccessful and that a re-calibration might be required. An example of a flat surface scan is shown in Figure 30 with the camera image shown on the left and the reconstructed surface with a fitted 2D plane shown on the right. The RMS error for this example was 339μm.

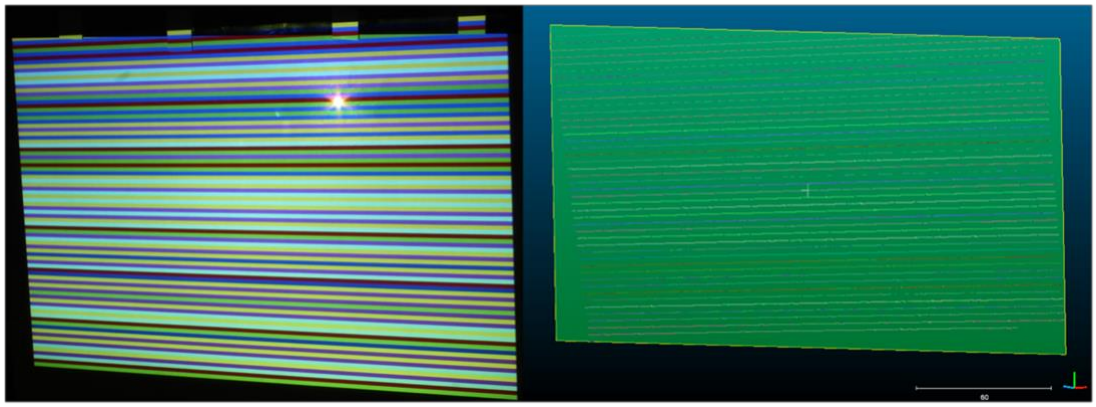


Figure 30: Camera image of a flat surface scan (left). Surface reconstruction (colour lines) and fitted 2D plane (green) (right). RMS error = 339μm.

For a 3D object reconstruction test, an object with known dimensions is required. Ideally, the object should have a shape and colour that mimics a real-world object, i.e. the human back. To simplify the initial testing, the first 3D object test was done on a completely white object. This was done to eliminate object colour as a variable and test for shape reconstruction accuracy only. A white surface reflects all light equally and therefore provides a best-case scenario for optical colour detection. The underside of a small plate was chosen for this initial test and is show in Figure 31. The shape of the plate provided a relatively smooth surface, but with a distinct, measurable feature, i.e. a circular base. The circular base of the plate was measured with Vernier callipers at Ø124.5mm. The text marking on the centre of the plate (shown in the figure) was deliberately included in the scan to see how the scanner would react to a surface blemish.

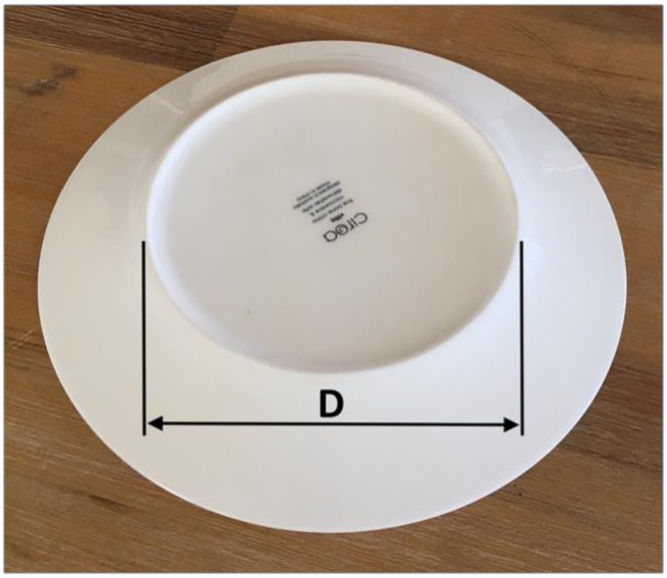


Figure 31: Small plate with a base rim diameter of Ø124.5mm.

The average base diameter of the reconstructed plate was measured to be within 1mm ($Ø_{reconstruction}$ = 123.5mm) of the original plate, a reconstruction error of less than 1%. In addition to this accuracy, repeated inter-session measurements did not produce a noticeable difference in object measurements. The reconstructed 3D point cloud of the plate is shown in Figure 32. As can be seen from the image, the text marking on the plate did have a negative effect on the reconstruction and some of the surface around the text could not be reconstructed.

To further validate the accuracy of the scanner, it was also tested on an object that better mimics the shape of the scapula under the skin. To simulate a scapula, a wooden object was constructed in the shape of a right-angled triangle with dimensions 115.4mm x 84.8mm x 20mm. A triangular shape was chosen because it resembles the simplified geometry of a scapula and it is easy to visually identify and orientate a triangular shape when covered by fabric.

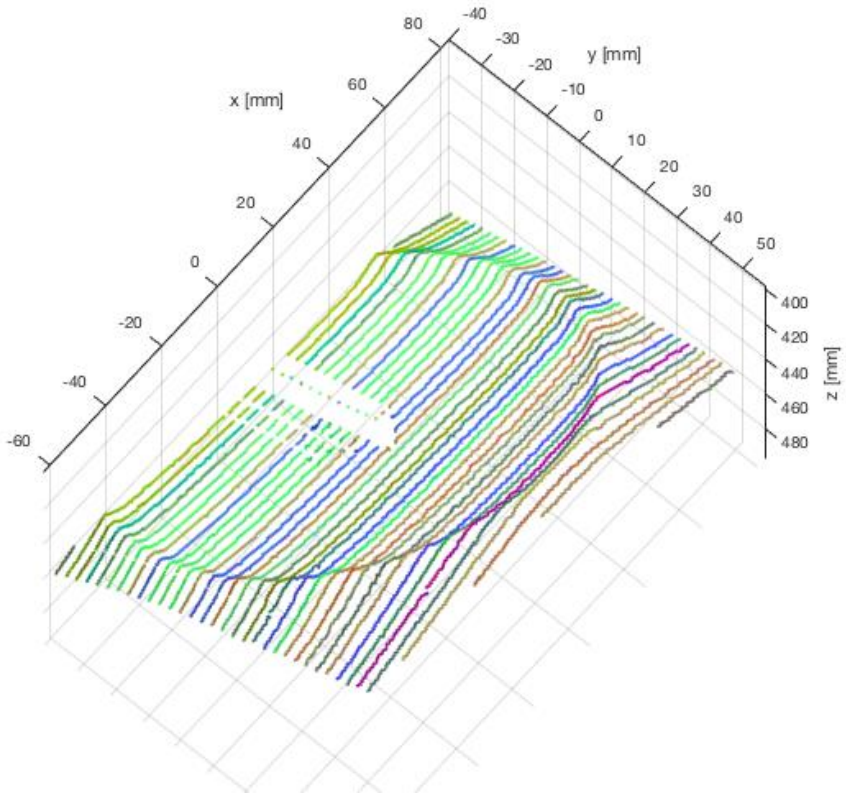


Figure 32: Small plate 3D reconstruction, base rim diameter of D = 123.5mm.

A piece of fabric, pulled over the wooden scapula was used to simulate the skin coverage. The goal for this second object test was two-fold, firstly to further validate the accuracy of the scanner on a smooth shape and secondly to determine if the scanner's selected resolution generated enough data points to capture a smooth shape without a distinct measurable feature. For example, when the wooden triangle is placed under a piece of fabric, a human observer with prior knowledge about the shape of the object can easily identify the location and the orientation of the triangle provided they have a high-resolution image of the fabric (Figure 33). As the image resolution drops, it becomes harder and harder to see the shape of the fabric and find the triangle, the resolution of the reconstruction is therefore critical.

Figure 33: Wooden scapula only (left) and covered with fabric (right).

To try and emulate varying skin-tones, the object was covered with different coloured fabrics. Using fabric to emulate human skin is an obvious simplification with several limitations, the main one being the difference in reflectance properties. Reflectance refers to the way light is reflected off an object and it can be influenced by several factors. A piece of fabric's reflectance properties are mainly determined by the type of material used to manufacture the fabric. For the human skin, there are many factors that influence the reflectance properties. For example, the presence (or absence) of perspiration and/or natural oils on the skin can cause it to appear shiny (or dry), and the level of pigmentation present in the skin cells can make it more or less absorbent to light. The way the skin interacts to light is much more complicated than fabric but, being keenly aware of these limitations, the test did still have some merit to it. The test was used to simulate the shape of the skin across the scapula as well as give a simple indication of how the scanner would react to lighter and darker objects. The test was done using white (as a control), grey, brown and black fabric coverings. The covered wooden scapula was scanned and reconstructed using the proposed system and it reacted well to the colour variance and reconstructions were possible even on the least reflective (black) fabric.

The resolution was deemed to be high enough and the shape of the triangle was still clearly visible on the reconstructions. In terms of the accuracy of the test, it was difficult to select the exact locations for the edges of the triangle on the reconstructed point cloud. This was mainly due to the fact that the fabric introduces a curved edge that makes exact edge selection difficult. That being said, the fabric measurements could be made repeatedly within 2-5mm of the wooden triangle's dimensions, measurements were confirmed by palpating the triangle's corners and marking them on the fabric surface. Figure 34 shows an example of a reconstruction. A mesh was fitted over the point cloud to make the surface more visible. Points A, B and C were palpated and measured on the mesh and the distance from A to B was D_{AB} = 120.3mm (4.9mm difference) and from B to C, D_{BC} = 87.7mm (2.9mm difference). These surface measurements included the fabric thickness of about 1mm, so the overall measurements were well within the design specifications. In conclusion, the accuracy of the 3D reconstructions proved to be adequate for making object and landmark measurements that are comparable to the accuracy of measurements by palpation only (2-5mm).

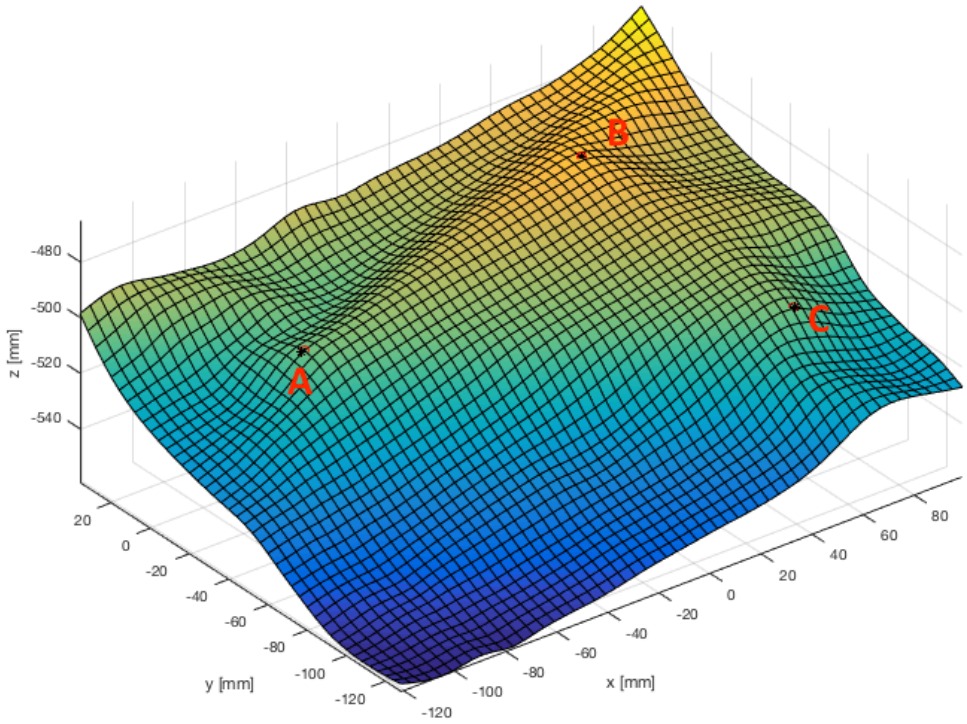


Figure 34: Reconstruction of wooden scapula covered by fabric. Selected triangle edge-points shown.

3.4.4 Test result summary and conclusion

Table 5 summarises the results of the laboratory testing compared to the initial design specification.

Table 5: Design specifications test results.

Req. ID	Requirement	Specification	Test results
01	Cost	< R13 500	< R10 000
02	Setup time	2 hours	< 2 hours
03	Calibration	Easy to do with minimal training	Easy to do with minimal training
04	Speed	1/70 seconds or faster	1/80 seconds
05	Scan area	600 cm^2 or more	1575 cm^2
06	Resolution	4000-6000 data points across the scapula	Approx. 34 000 data points across the scapula
07	Accuracy	2-5mm or better	Within 2-5mm
08	Repeatability	20% deviation	<400μm which is 20% for a 2mm accuracy (lower limit)
09	Light level	No design specification	0-100 lux

The laboratory simulations showed that the designed system performed better than the initial design specifications and no major changes to the system were required. In addition to the design specification validation, the laboratory tests proved valuable in pointing out a few important considerations.

Firstly, the importance of the ambient light's influence on the scan results, which was not known prior to testing, became evident. The testing showed that the system performance can be negatively affected when too much ambient light is present. This effect can however be managed effectively using the ISO and aperture control settings of the camera. A light meter was used to monitor the ambient light levels around the scanner during the test phase to quantify the ideal operating range. It was established that the performance of the system will remain unaffected in the range of 0-100 lux. Requirement 09 was therefore added to the system specifications (Table 5).

Secondly, it was also shown that the presence of large blemishes on the surface can make reconstruction difficult. This is due to the fact that a blemish can create a detectable edge on the camera image. In order to make the scanner more robust against surface discolouration the edge detection algorithm was updated and improved to try and eliminate false positives on edges that are not directly caused by the deBruijn colour pattern.

Finally, it was difficult to repeatedly select the edges (or landmarks) of the triangle on the reconstructed point cloud using only the reconstructed surface data and the knowledge of the underlying shape. Interestingly, it is easy to intuitively point out (locate within 10mm) the landmarks of the triangle by merely looking at the fabric surface, but when tasked to select their exact locations (within 1mm, for example), it becomes increasingly difficult due to ambiguity introduced by the curved shape of surface. Selecting these landmarks mathematically by means of calculating some mathematical calculation would be very helpful.

In conclusion, the laboratory testing phase showed that a 3D structured light scanner can be used as a cost-effective way to create a surface map of objects that resemble the human back. The accuracy and resolution of the surface map proved to be adequate for shape and landmark identification, but some work is required to make landmark detection easier.

Section II: Clinical Methodology

Thus far, it has been shown that a surface map approach can be used to evaluate the scapula. Even though current surface mapping approaches still make use of marker-based technology, several motivating factors prompted the development of an optical, marker-less, surface mapping tool. The main factors were:

- the fact that current clinical methods for evaluating scapular dyskinesis make use of visual observations (which is in essence an optical method),
- the need to improve on some of the limitations associated with marker-based methods, and
- the desire to explore novel methods of evaluating the scapula and expand research on marker-less methods.

This tool employs 3D structured light technology to create a high-resolution surface map of the human back using only a single photograph. An optical surface mapping approach has significant advantages over a marker based approach in the following areas:

- no time-intensive manual marker placements are required,
- a single system calibration is required instead of subject-specific calibrations,
- a precise surface map can be created repeatedly and on different days, eliminating errors introduced by manual marker re-placements, and
- no costly equipment is required compared to marker-based methods.

A marker-less surface mapping approach does, however, present some challenges. The main challenge is to identify and accurately locate the landmarks using only the surface topology. Even current marker-based surface mapping studies, like Mattson *et al.* and Charbonnier *et al.* (Table 2), always include a marker on at least one major scapular landmark like the acromion. This marker is then used as the starting point for any surface-based analysis. So, even though they employed a surface mapping approach, their results were still bound, to a large extent, by the limitations of a marker-based method. This project, however, aims to employ a pure surface mapping approach in an attempt to completely eliminate the need for markers and explore the limits of what is possible using only surface data. This section outlines

the clinical methodology that was followed to evaluate the diagnostic potential of our optical, marker-less, surface mapping tool.

3.5 Diagnostic design

To develop a useful marker-less measurement tool, the tool must at be able to assist a clinician with either making a diagnosis and/or monitoring post-treatment results. To do this, such a tool must ideally be able to add value in one or more of the following ways:

- measure a variable that is otherwise unmeasurable,
- offer a more efficient or novel method for measuring a variable,
- provide a new way of visualising the clinical problem (either through novel data presentation or through new medical imaging capabilities).

In developing a measurent tool for scapular dyskinesis, it is clear that the position and pose of the scapula is important. Any alterations in normal resting position and motion of the scapula can be observed visually, but measuring these alterations are difficult, even with a marker-based approach. During a visual evaluation, a clinician may refer to a medial border and/or inferior angle prominence. Scapular landmark prominence can be an indication of an altered scapular resting position because the tilted or rotated scapula will create a protrusion of the skin. These protrusions or prominences may also be visible, and are sometimes even more visible, during arm movements. A prominent scapula is also easier to detect when the prominence is asymmetrically present, this allows for a right-left comparison which can highlight any differences due to protrusions.

Currently, marker based methods can provide an estimate of certain scapular landmark locations in varying degrees of accuracy, with the acromion considered as the most accurate and the inferior angle as the most inaccurate (due to STAs) (Lempereur et al., 2014). These landmark locations can be used, to a certain degree, to infer a scapular location and pose relative to the sternum. Although, the location and pose of the scapula is important and extensively used in biomechanical analysis of the shoulder, it is hard to determine the degree of scapular misalignment using only the sternum and the scapula's location and pose. The shape of the anterior thorax plays an important role and cannot be measured using only markers. The thorax essentially determines a person's unique scapular resting position.

A clinician's trained eye can detect a misaligned scapula through visual observation and complex anatomical assumptions about the thorax and scapular position based on their training and prior experience. They use this skill to determine if a scapula is indeed misaligned and may diagnose scapular dyskinesis based on their observation. The author speculates that when a clinician makes a diagnosis about scapular dyskinesis, he or she is less concerned with the exact locations of the scapular landmarks and more concerned with the visual cues created by the protrusions of the scapula. These visual cues provide the information used to make a diagnosis but there are no methods currently available to measure them. In developing a measurement tool, the aim was to try and mimic the clinician's visual 'methods of analysis' to capture and to quantify/measure these visual cues.

3.5.1 Surface curvature analysis

In an effort to develop a measurement tool that can be used to evaluate scapular dyskinesis, a surface curvature (SC) analysis method is proposed to extract and measure the surface protrusions caused by the scapula. A protruding scapula deforms the skin into a curve that is visually detectable, and this curve or protrusion represents a portion of the scapula that has lifted from the rib cage. Figure 35 shows the left and right scapulae protruding at the IA.

Figure 35: Protruding scapula (Metzger, 2013).

It is proposed that the amount of protrusion can measured by considering the mean curvature of the skin at the point of protrusion. This concept is visually illustrated in Figure 36 where a high mean curvature (shown only in 2D) is caused by a large protrusion of the bone from the rib cage. The concept was developed in response to the problem of accurately selecting landmarks on a curved surface that was encountered during laboratory testing.

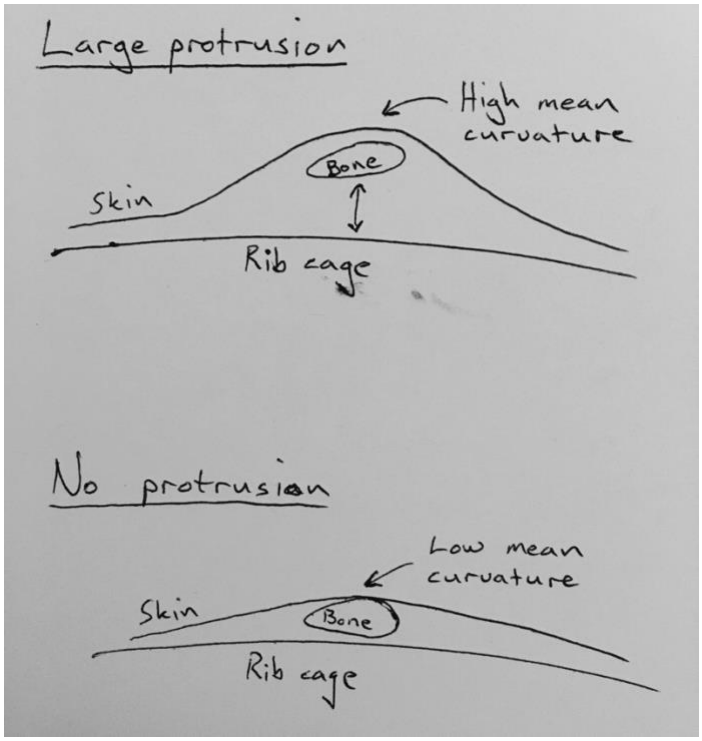


Figure 36: Conceptual drawing of protrusion linked to mean curvature.

This mean curvature can be measured using differential geometry. At any point on a curved surface, the amount of surface bending (or curving) can be measured by computing the Gaussian or mean curvature. The mean curvature represents the extrinsic measure and has proven to be most useful in this application. Mean curvature (H) is an extrinsic measurement of the SC, defined as:

$$H = \frac{(\kappa_1 + \kappa_1)}{2} \tag{10}$$

where κ_1 and κ_2 are principle curvatures. The principle curvatures are defined as the maximum and minimum bending of a surface at the point of interest. Mean curvature is measured in length^{-1}. An in-depth mathematical description of mean SC is not necessary here but an interested reader is referred to an online resource (Weisstein, 2015).

A Matlab implementation of the SC by Claxton (2006) was used to calculate the mean curvature. In order to visualise the curvature data, it was superimposed onto the 3D topographical surface as a heat map. The practical effect of a mean SC calculation can be seen in Figure 37. The image was generated using the same data as for Figure 34 and the wooden scapula corners are marked with black dots. Figure 38 presents a side-to-side comparison of the before and after results.

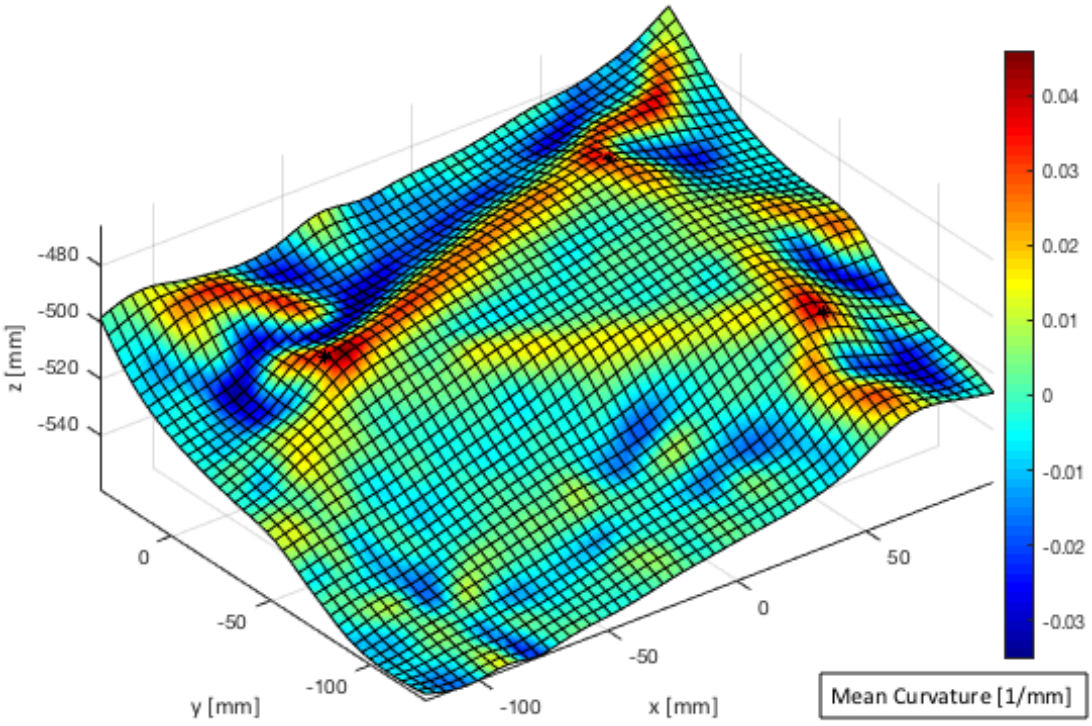


Figure 37: Wooden scapula covered with fabric with mean curvature shown.

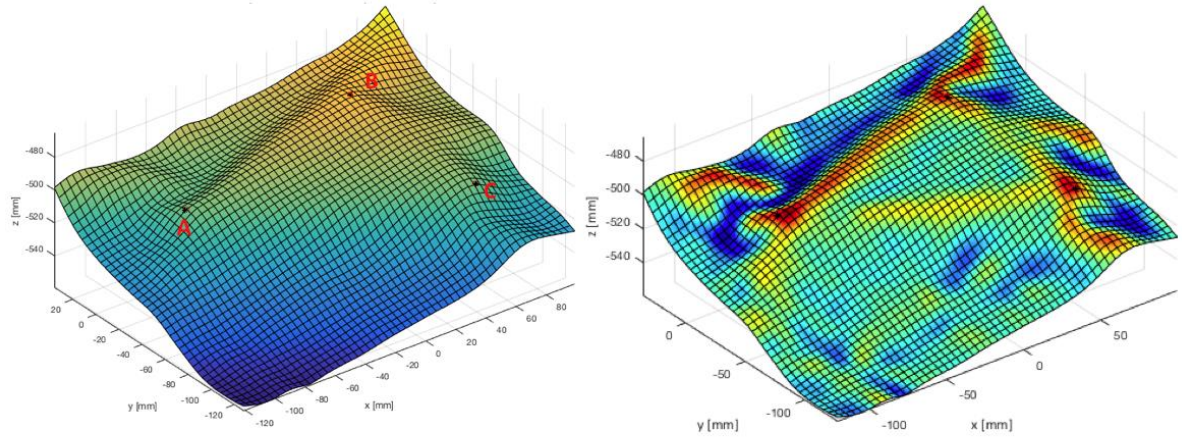


Figure 38: Comparison, before (left) and after (right) SC calculation.

A positive SC value (shown in red) represents a convex surface (or a peak) while a negative value (blue) represents a concave surface (or a valley). A SC analysis is a mathematical way of calculating where the surface bends the most. It is assumed that the maximum positive curvatures can therefore be found at the edges and corners of the triangle. It is clear from the image that this is in fact the case. Note that a few other peaks and valleys produced by the folds of the fabric are also visible.

In essence, when presented with a surface that is shaped by an underlying rigid structure, a SC analysis allows us to select the points of maximum protrusion and measure the amount of curvature present. It is therefore proposed that a similar surface analysis can be used to analyse the skin across the human back, which is also curved due to protrusions caused by a rigid structure, i.e. the scapula.

3.6 Clinical testing

The purpose of the clinical testing was twofold, firstly, to test the designed tool on important variables that could not be simulated in a laboratory. Some of these variables included:

- the influence of soft-tissue coverage over the scapula,
- the effects of movements caused by respiratory and cardiac activity on the results, and
- the variability in the reflectance properties of human skin.

Secondly, to determine the diagnostic potential of the tool by evaluating it according to the proposed diagnostic design. This section outlines the experimental procedure that was followed.

3.6.1 Characteristics of the study population

In a systematic review of the similar studies that involved marker based scapular motion analysis methods, Lempereur *et al*. (2014) showed that population sample size varied between five and 26 (avg. 10) across 19 different studies. In another validation study, with objectives similar to this study, researchers used 14 participants (Mattson et al., 2012).

In order for the results of the proposed hypothesis to be statistically significant the statistical power P, equal to (1-β) with β being the statistical probability of accepting a false null hypothesis, should be sufficiently large for the false null hypothesis to be rejected. To prevent this error, the number of participants should accurately represent the population and ensure that the results are not influenced by the random sampling error. A typical power value is 80%, with $\beta = 0.2$. The significance value α should also ensure that the hypothesis test has a sufficiently small probability of accepting a false hypothesis. A typical value for α is 0.05. With

a power value of 90% and α = 0.05 a sample size of 11 was estimated for this study using a mean measurement accuracy of ± 2mm for the 3D locations of the scapular landmarks.

A total of 12 participants were recruited to take part in the study and combined Body Mass Index (BMI) and hip-to-waist (H-W) ratios were recorded. Hip-to-waist ratio is calculated as the waist circumference divided by the hip circumference (H-W ratio = Waist_C ÷ Hip_C). BMI is calculated as the body mass divided by the square of the body height (BMI = Mass ÷ Height^2). The categories for H-W ratio and BMI according to the World Health Organisation (2008)(WHO) are shown in Table 6:

Table 6: BMI and hip-to-waist ratios for men and women.

		Women	**Men**
BMI	Normal	18.5 < 25	
	Overweight	25 and higher	
H-W ratio	Normal	<0.85	<0.90
	Overweight	0.8 and higher	0.9 and higher

A population of adults above the age of 18 was chosen for this study with the following inclusion criteria:

- Only participants fluent in English will be enrolled in the study.
- Should be able to lift their arms above their heads.

Prospective participants were excluded based on the following exclusion criteria:

- If they have any pain, “pins and needles” or numbness in any part of their arms, shoulders, neck or back (this is due to the sustained positions needed for this stage of the tool development testing).
- Had any previous surgical procedures done to the glenohumeral joint or scapula.

3.6.2 Ethical considerations

Prior to the initiation of the study ethics approval (HREC REF: 414/2-17) was granted by the University of Cape Town Faculty of Health Sciences Human Research Ethics Committee. Participants were required to read and sign an informed consent form prior to testing. The

approved Informed Consent Form and Research Protocol can be found in Appendix II: Informed Consent Form.

3.6.3 Test procedure

The main equipment used to conduct the research consisted of the following:

a) A structured light 3D scanner that consists of a standard projector, an SLR camera and a laptop computer to run the scanner software.
b) Inclinometer (Digi-Pas DWL-80E) to measure arm angles.
c) Scale and tape measure for BMI and H-W ratio measurements.
d) Light level meter (Goldilux GAL-2H) to ensure that the ambient light levels were within the acceptable range.

Participants were subjected to the following procedure:

Step 1 – Preparation

- Male participants were asked to remove their shirts, and female participants were asked to wear halter tops during the study to allow visualization of the posterior thorax.
- The following measurements were made:
    - Mass and height to calculate the BMI = Mass ÷ Height^2.
    - Hip and waist circumferences to calculate the H-W ratio = waist circumference ÷ hip circumference.
- Ambient light levels were measured.
- The participant was asked to stand in front of the scanner with feet shoulder width apart and aligned with a mark on the floor (Figure 39).

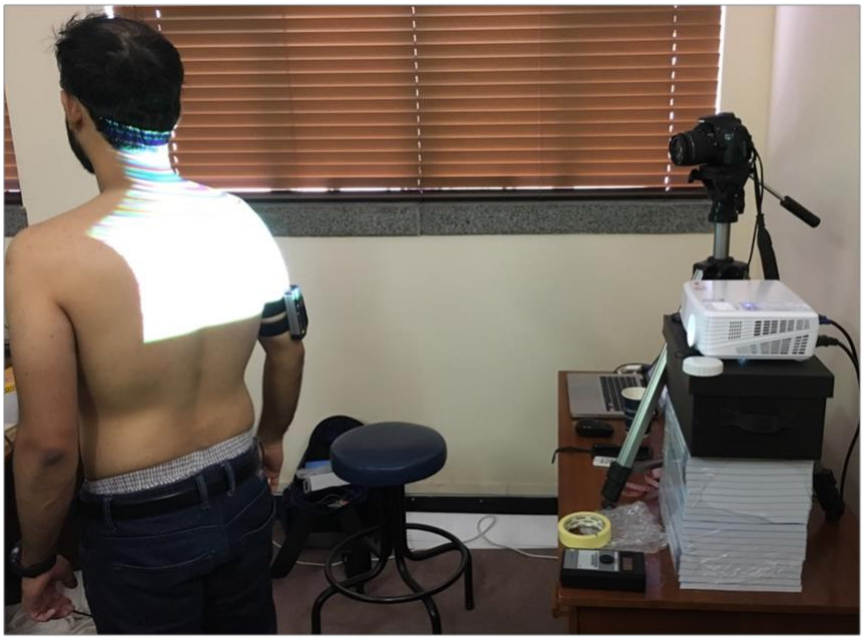


Figure 39: Typical scan setup with participant shown in front of the scanner.

Step 2 – Preparation and scapula template scan

- The participant was asked to place their arm behind their back, bend the elbow and protrude the shoulder backwards. This position makes the scapula protrude and become maximally visibility (Figure 40).
- The participant was asked if they experienced any pain. If any pain was present, the testing was stopped, and the participant was removed from the study.
- The position was held for a few seconds while the patient was scanned using the scanner.
- Total time for step 1 was about five minutes.

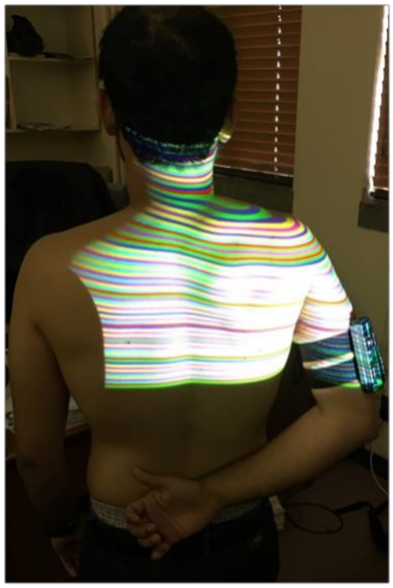


Figure 40: Participant with arm behind back to create a protruding scapula.

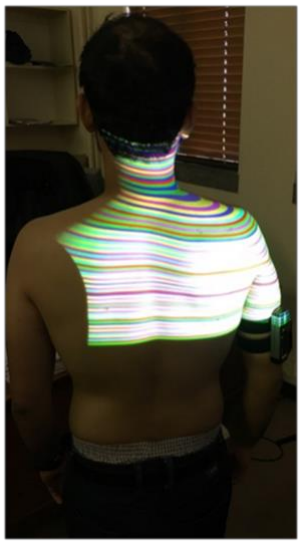
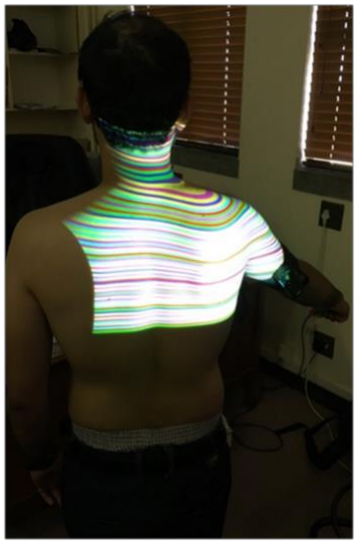
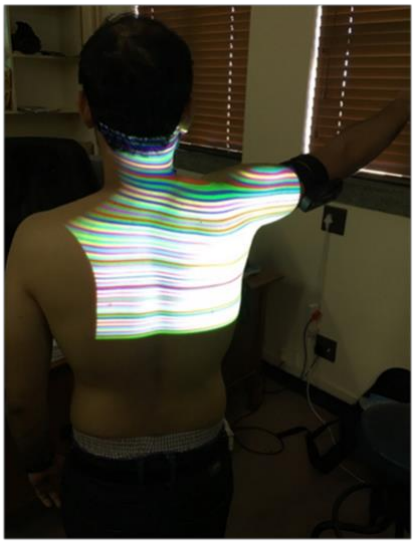
Step 3 – Primary testing: marking scapular landmarks at three arm positions

The participants were asked to place their right arm in three positions that are commonly associated with assessing the scapula (Kibler & McMullen, 2003).

The positions were (as shown in Table 7):

a) arm at side (defined as 0^o),
b) arm abducted 45^o from the 0^o position (referred to as 45^o abduction)
c) arm abducted 90^o from the 0^o position (referred to as 90^o abduction)

Table 7: Arm abduction positions with inclinometer visible on the right arm.

Arm 0°	Arm at 45^o abduction	Arm at 90^o abduction
		

The following procedure was followed for **each of the three arm positions**:

- The participant was asked to move their arm to the appropriate position with an inclinometer placed on the upper arm to measure the arm angle
- The participant's back was scanned with the arm held stationary. Each scan covered the following landmarks of the scapula (Uhl et al., 2009):
    - inferior angle (IA),
    - medial border (MB) at the root of the spine of the scapula,
    - superior angle (SA) and
    - acromion angle of the scapula (AA).

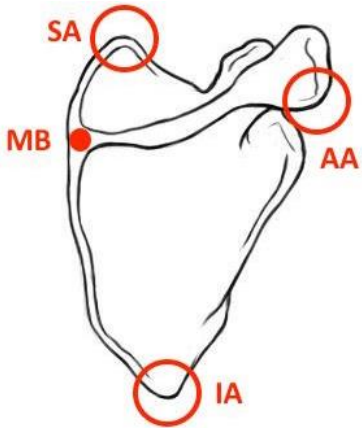


Figure 41: Main scapular landmarks shown, image adapted from (LoveOfDrawing, 2016).

Figure 42: A prominent right scapula with inferior angle (IA) and medial border (MB) clearly shown (Uhl et al., 2009).

- Next, while the arm was kept still, the scapula landmarks (IA, MB, SA and AA) were palpated and marked on the skin using a marker, and another scanner image was taken.
- Before moving on the next arm position the participant was given a few minutes to rest his/her arm.
- Total time for step 2 was approximately 20 minutes.

Step 4 – Conclusion

- Participant testing was finished in a single session and after the above protocol was completed the participant was notified that the test had been completed.

3.6.4 Data collection and safety

Only information described in the test protocol was collected and used for this study only. Data collection included photographs and physical measurements of participants. Care was taken to keep all photos anonymous by blurring out faces where necessary. All data collection and storage was done by the author and stored on a password protected personal computer.

4 Results and discussion of clinical tests

The aim of this thesis was to develop a measurement tool with quantitative capabilities and the results were therefore expected to be primarily quantitative in nature. However, due to the explorative nature of the project the results also contained a qualitative component in the form of a novel data visualisation technique for analysing the scapula under the skin.

The chapter starts with an example of the general scanning process followed by a summary of the BMI and H-W ratio measurements for all participants and a table with all the SC data results. This is followed by a discussion of the results for the following specific cases of an individual with:

- a prominent scapula,
- a less prominent scapula,
- a very muscular back, and
- a thick layer of subcutaneous fat.

The chapter concludes with a discussion about some of the important clinical variables that were tested for and the diagnostic potential of the scanner.

4.1 Scanning process

The scanning process that enables a projected image and a camera image to be converted into a 3D surface map is shown in Figure 43 and described below.

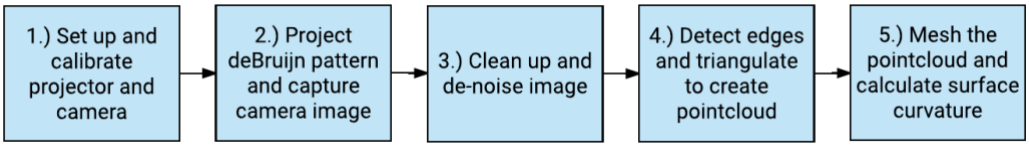


Figure 43: The scanning process.

4.1.1 Process steps

The results from participant 11(F), using the case with the arm at 0° abduction, is used to illustrate the process.

1. Set up and calibrate the projector and camera.

2. Project the deBruijn pattern onto the participant's back, the captured camera image is shown in Figure 44. Three small black marks are visible on the scapular landmarks that were palpated: the inferior angle (IA), medial border at the root of the spine (MB) and the acromion angle (AA).

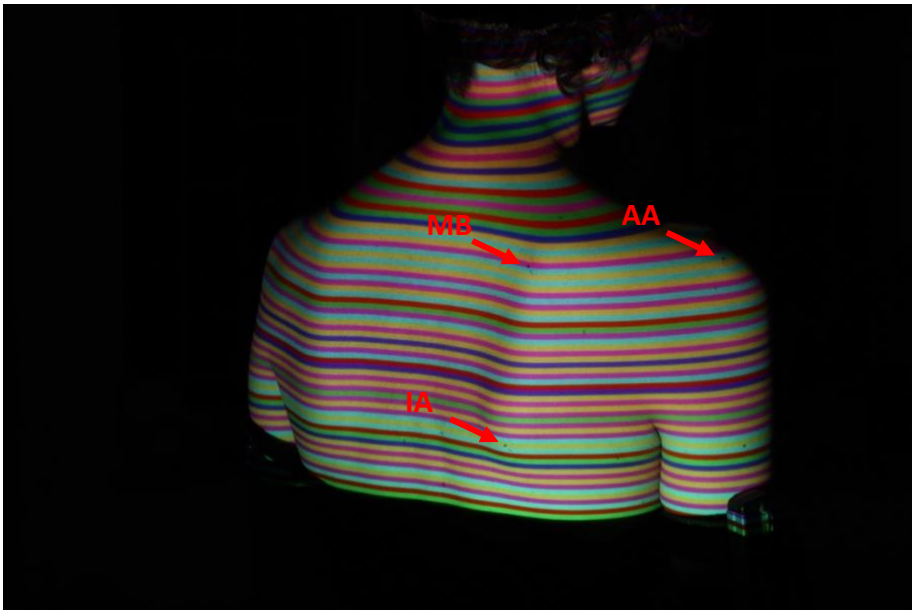


Figure 44: Camera image of participant 11(F), arm at 0° abduction.

3. Crop a selection from the image for image processing and noise removal. Only a section of the image is used, dark areas are removed to avoid unnecessary image processing to improve performance. The cropped section inside the original image is show in Figure 45.

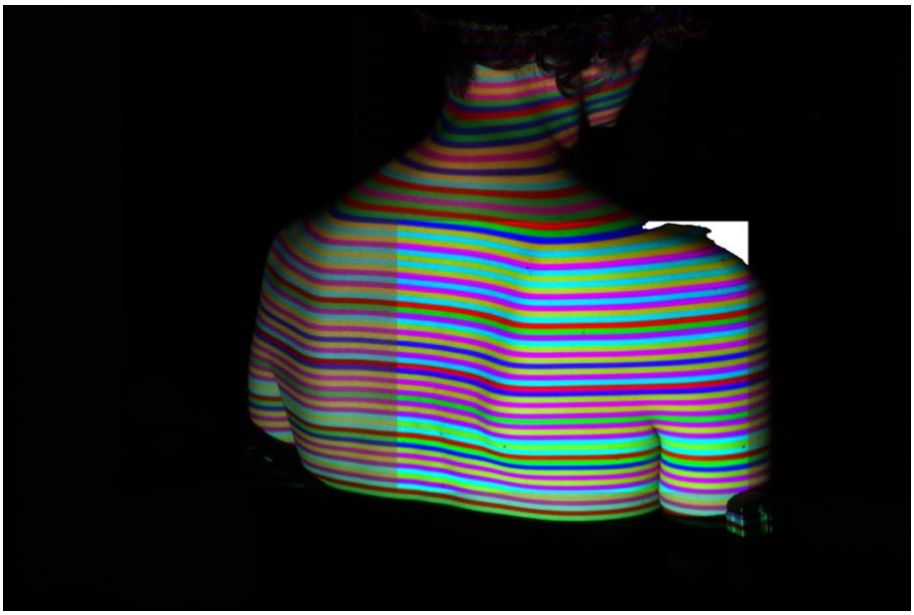


Figure 45: Selection from camera image subjected to colour correction and a de-noise filter and dark sections removed.

4. Matlab is used to do the edge detection, edge matching and triangulation to produce the 3D point cloud of the cropped section. The point cloud is subjected to some post-processing which involves the removal of outlier and noisy points. The result is shown in Figure 46. Note that the section near the armpit (with the inclinometer) was removed as it was too noisy.

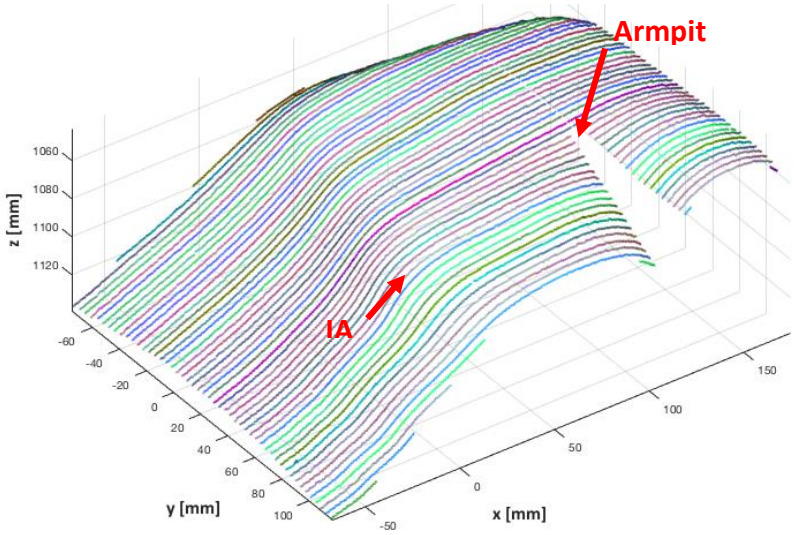


Figure 46: Point cloud surface map data of participant 11(F), arm at 0° abduction.

5. Finally, a mesh is fitted to the point cloud and the mean SC is calculated and superimposed onto the 3D meshed surface. In addition to this, the small black marks on skin, that indicate the location of the palpated landmarks (IA, MB and AA), are also on the 3D surface. The result is shown in Figure 47.

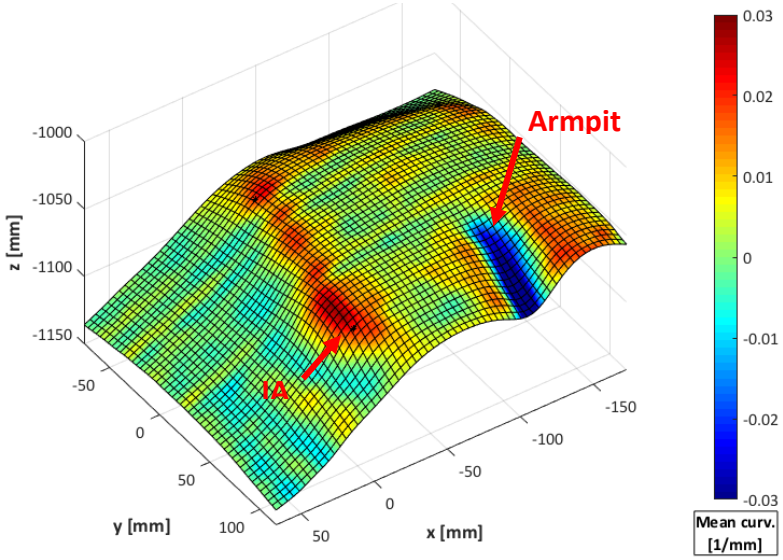


Figure 47: SC results of participant 11(F), arm at 0° abduction.

4.2 Results

4.2.1 Participant BMI and hip-to-waist ratios

A total of 12 participants (eight male, four female) took part in the study with an average BMI of 22.05 ± 2.89 and H-W ratio 0.80 ± 0.05. Three participants had above normal BMIs but their H-W ratios were normal, hence they were classified as normal (as per Table 6). The results for all participants are summarised in Figure 48 with normal ranges for BMI and H-W ratios shown in orange (note that the normal H-W ratio is 0.85 for females). Male participants are placed from one to eight and female participants from nine to 12 according to ascending BMI values.

4.2.2 Surface curvature data

The BMI and H-W ratio data is followed by a tabulated summary of the SC data for all participants in Table 8. The results for arm abduction to 0°, 45° and 90° is shown in the same order as the BMI and H-W ratio results to allow for easy referencing. The palpated scapula landmarks are marked with black markers and the participant's spine is easily identifiable as a blue valley to the left of the scapula. For participant 10, the surface map could not be reconstructed around the AA for the 45° and 90° abduction positions, hence the AA marker is not included for these two images.

In order to ensure that the SC results were consistent and comparable, the measurement needed to be standardised. All the final results were analysed, and it was found that a mean curvature range of ±0.03mm^{-1} provided good results that included most of the variability across the study population. For example, the participant with the most extreme scapula protrusion produced peak SC results in the order of 0.03mm^{-1} while a flat unobtrusive scapula produced SC values between 0 and 0.01mm^{-1}. It was decided to also include negative curvature values (valleys) up to -0.03mm^{-1} to allow the spine and any valleys near the scapular landmarks to be clearly visible.

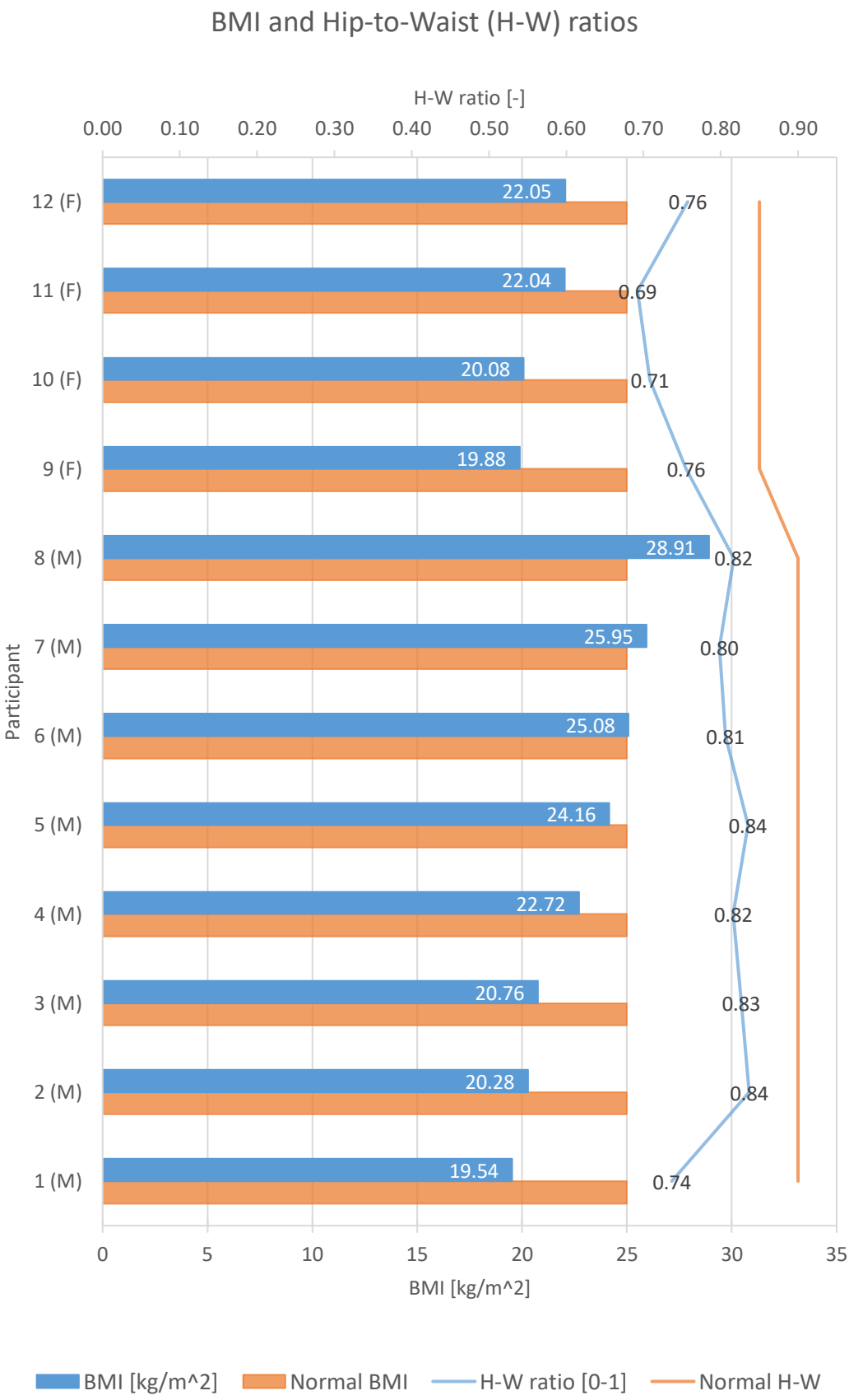
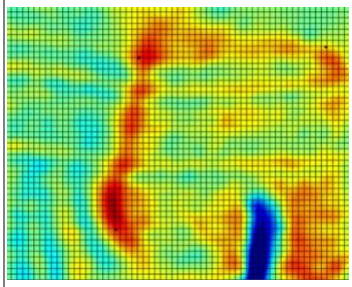
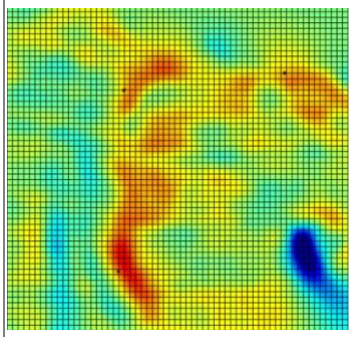
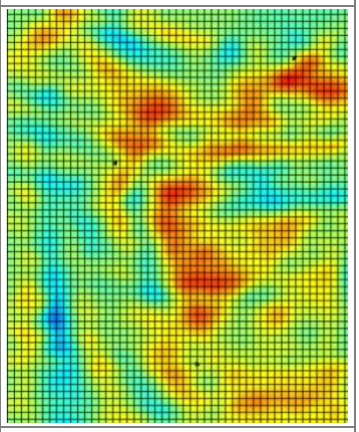
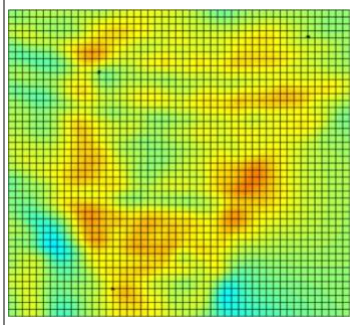
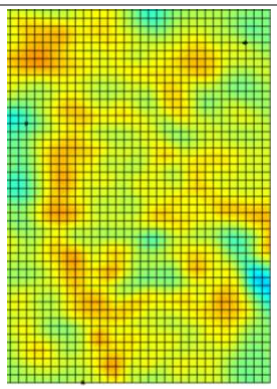
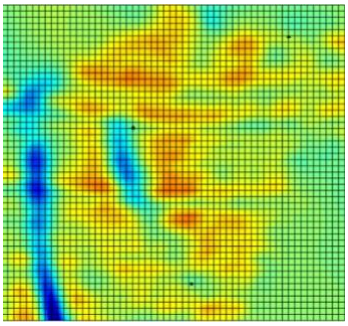
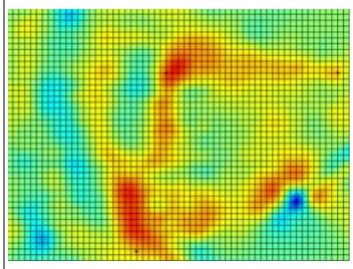
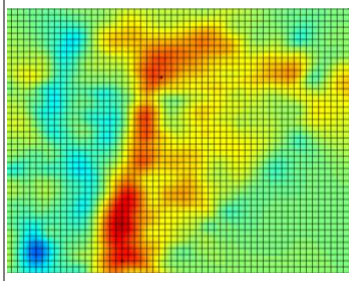
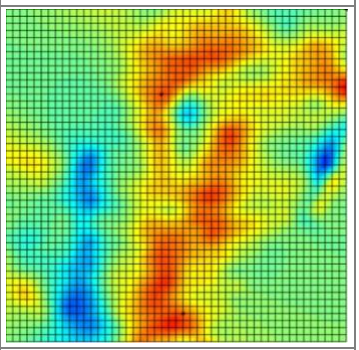
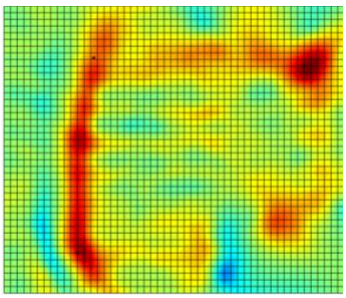
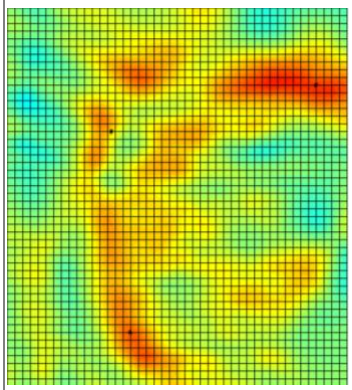
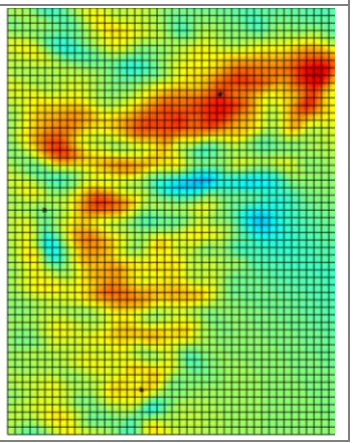
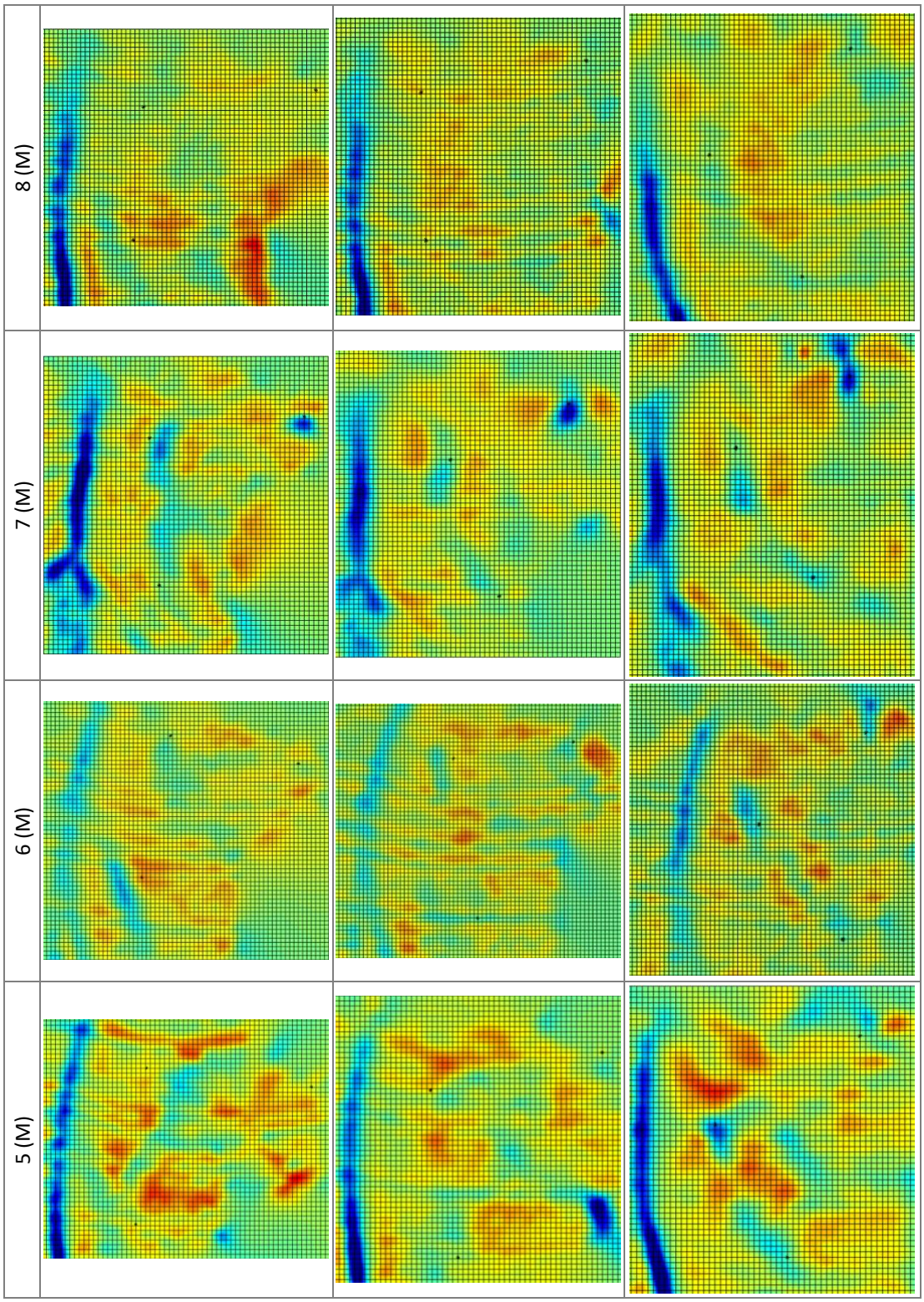
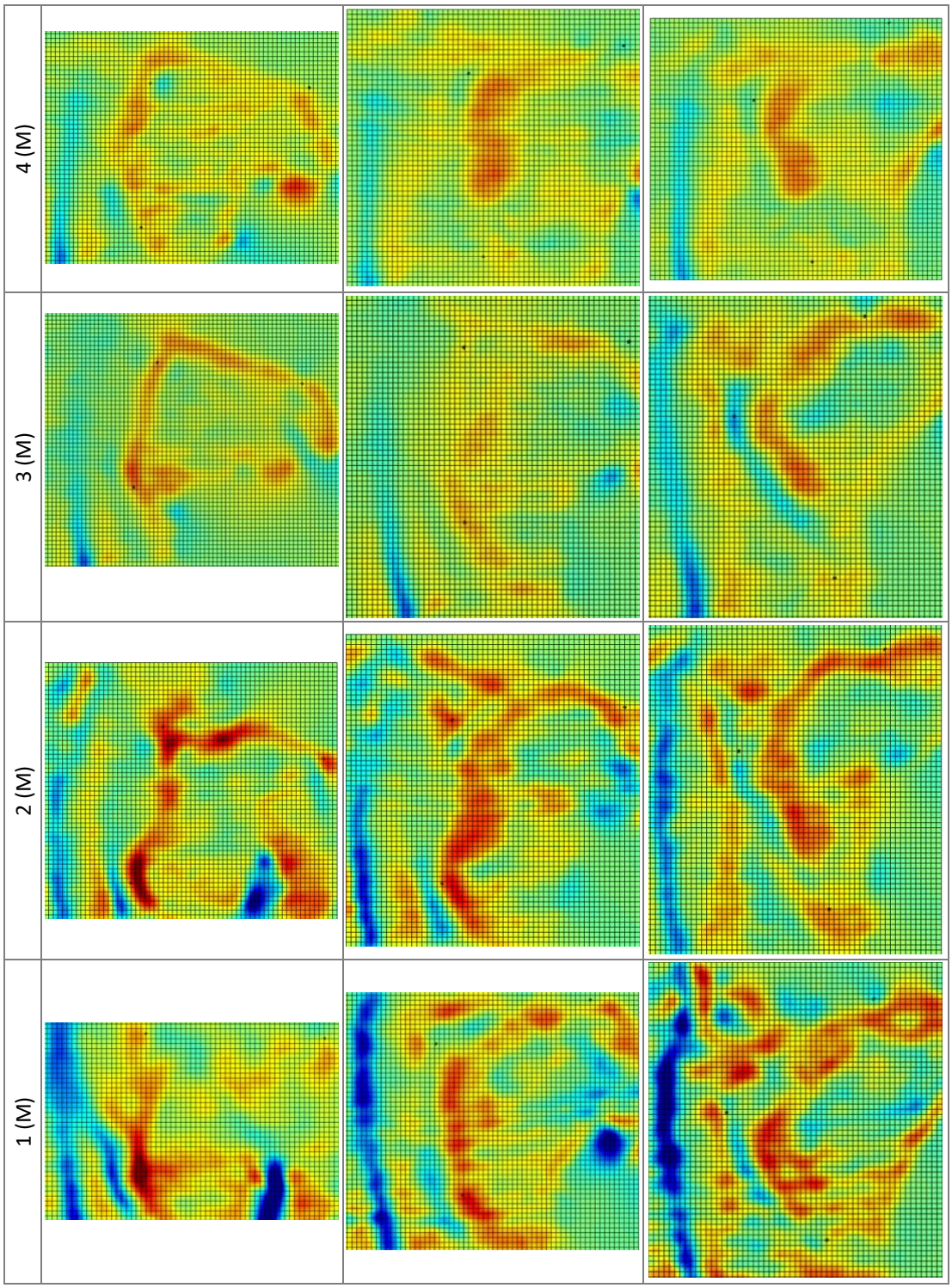


Figure 48: BMI and hip-to-waist ratios for all participants.

Table 8: SC results for all participants.

*L	-0.03 -0.02 -0.01 0 0.01 0.02 0.03		Mean curvature [1/mm]
#	0° abduction	45° abduction	90° abduction
12 (F)			
11 (F)			
10 (F)			
9 (F)			


8 (M)
7 (M)
6 (M)
5 (M)



*L = legend.

4.2.3 Locating the scapula

One goal of the clinical study was to determine if the surface mapping approach could be used to determine the location of the scapula under the skin without the need for physical markers. To achieve this, a SC analysis approach was proposed to augment the surface map data and highlight the skin protrusion created by the underlying structure of the scapula. To test this, the scapula landmarks were palpated and marked on the skin and captured using the camera. This camera image was used to create a 3D surface map of the scapula and calculate the 3D locations of the palpated landmarks. In doing so, the 3D landmark locations could then be compared with the features (peaks and valleys) created using the SC analysis data.

As expected, the SC analysis approach proved to be best suited for individuals with a low BMI, and specifically those with a clearly visible scapula. The SC features identified the location of the scapula and correlated with the palpated landmark locations. For examples of individuals with low BMIs, see the results for participants 1, 2, 3, 9, 10 and 12 in Table 8. Also note the pronounced SC features from the results of participants 1, 2, 9 and 12, who presented with prominent scapulae.

The SC approach proved to be the least effective for individuals with a less prominent scapula (participants 4 and 11), or those with a high BMI or a thicker subcutaneous fat layer (participants 6 and 8). This was expected as a less prominent scapula would create no protrusions and a high BMI individual would have a thicker layer of subcutaneous fat which would obstruct the scapula. Three participants presented with very muscular backs (participants 5, 6 and 7) and the SC results produced interesting results. Instead of creating peaks at the landmarks locations, valleys appeared at some of the landmarks, specifically the MB and AA. These valleys provided a good estimation of the scapula at rest as well as with abduction to 45° and 90°.

4.3 Specific cases

Following the summary of all the results, a few specific cases will be highlighted and discussed in order to put the results into perspective and to show that scapular measurements were possible in various situations. Firstly, a participant with a prominent scapula, followed by one

where the scapula was less prominent. Next, a participant with a very muscular back and lastly, an individual who had a thick layer of subcutaneous fat are discussed. The camera image for the resting position (arm at 0°) is given for each case, followed by the SC data for the three arm positions (0°, 45° and 90° abduction to the side). The SC data is presented as 2D heat maps with the IA, MB and AA landmarks shown (black markers).

4.3.1 Prominent scapula

This participant (No. 2) presented a prominent scapula where the inferior angle and lower part of the medial border clearly visible. Figure 49 shows the camera image with the arm at rest. The IA, MB and AA landmarks were palpated and marked on the skin, these are also shown on the image. The X-Y view of the topographical map with the SC data for the arm at rest is shown in Figure 50.

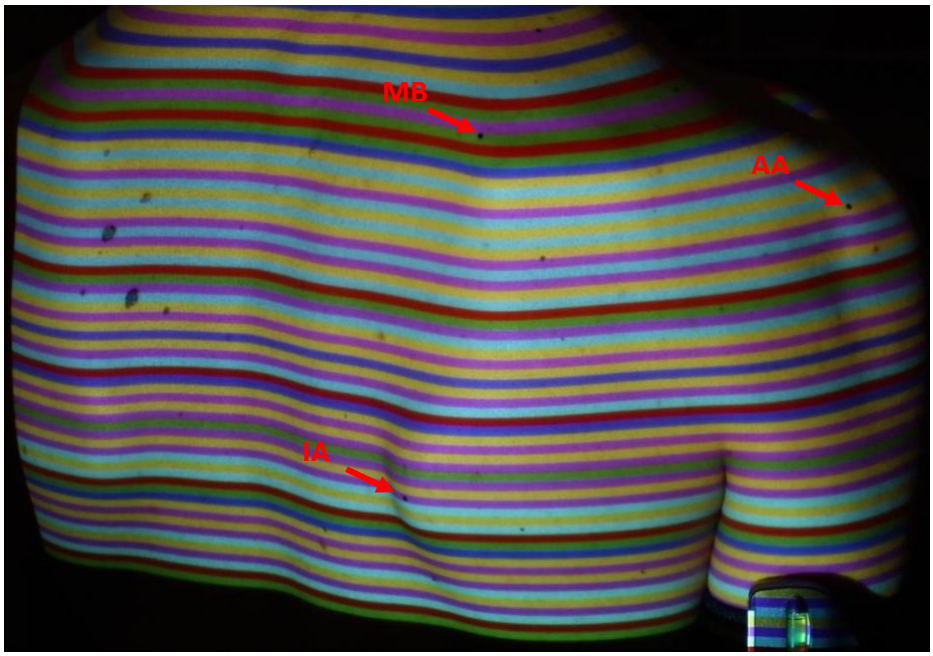


Figure 49: Participant 2 (M), camera image at 0° abduction.

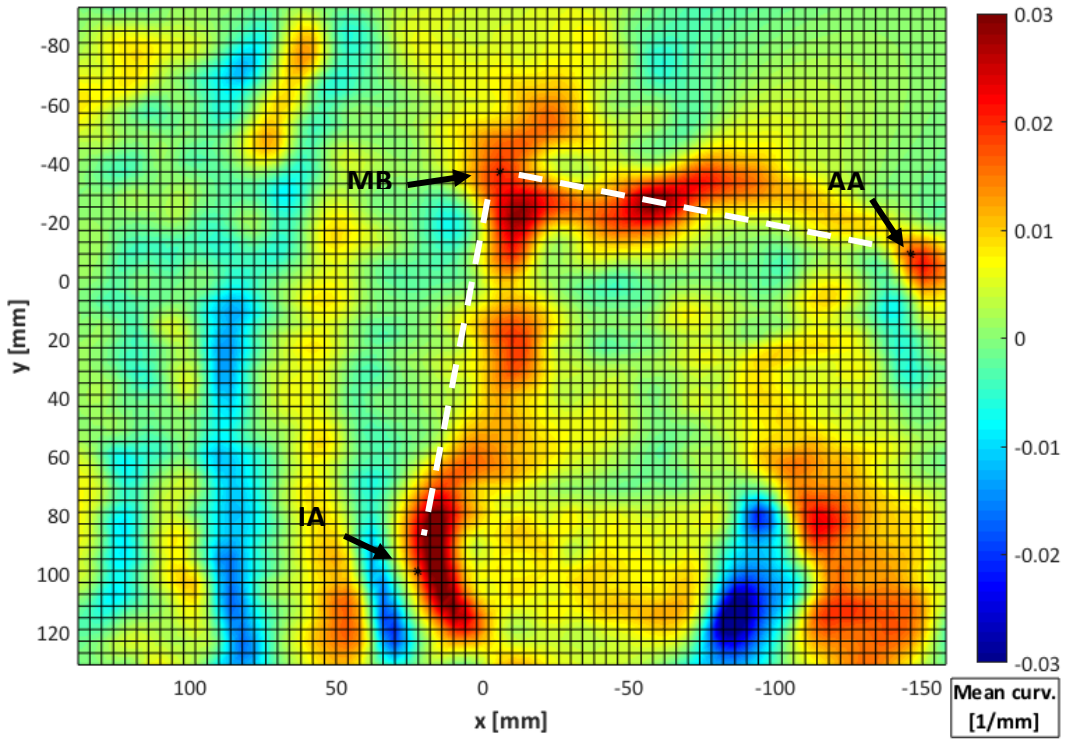


Figure 50: Participant 2 (M), 0° abduction (white line indicates scapula position).

The IA, MB and AA landmarks are also marked on the SC map using small black marks. Two lines are drawn between the three landmarks to indicate the estimated position of the medial border and spine of the scapula. Note how the red areas of high positive curvature correlates with the palpated landmark positions. Also note the blue valley on the left that correlates to the position of the spine.

Figure 51 and Figure 52 show the SC maps for the scapula with the arm abducted to 45° and 90° respectively.

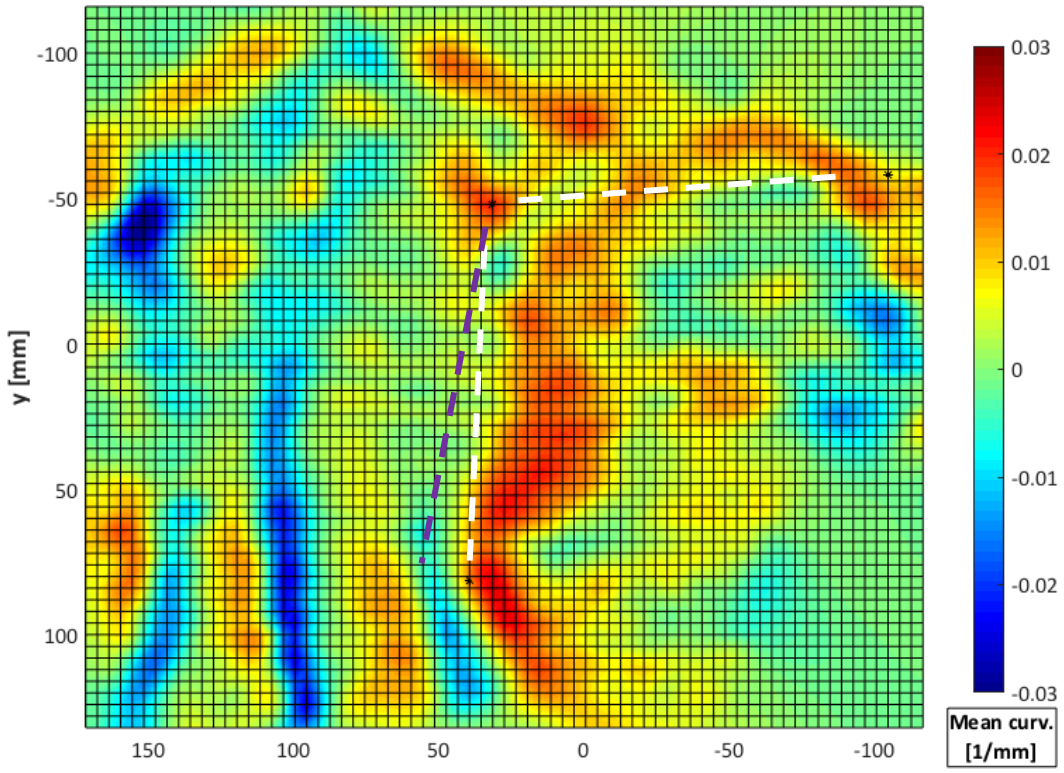


Figure 51: Participant 2 (M), 45° abduction (white line indicates scapula position at 45°, purple line indicates the position at 0°).

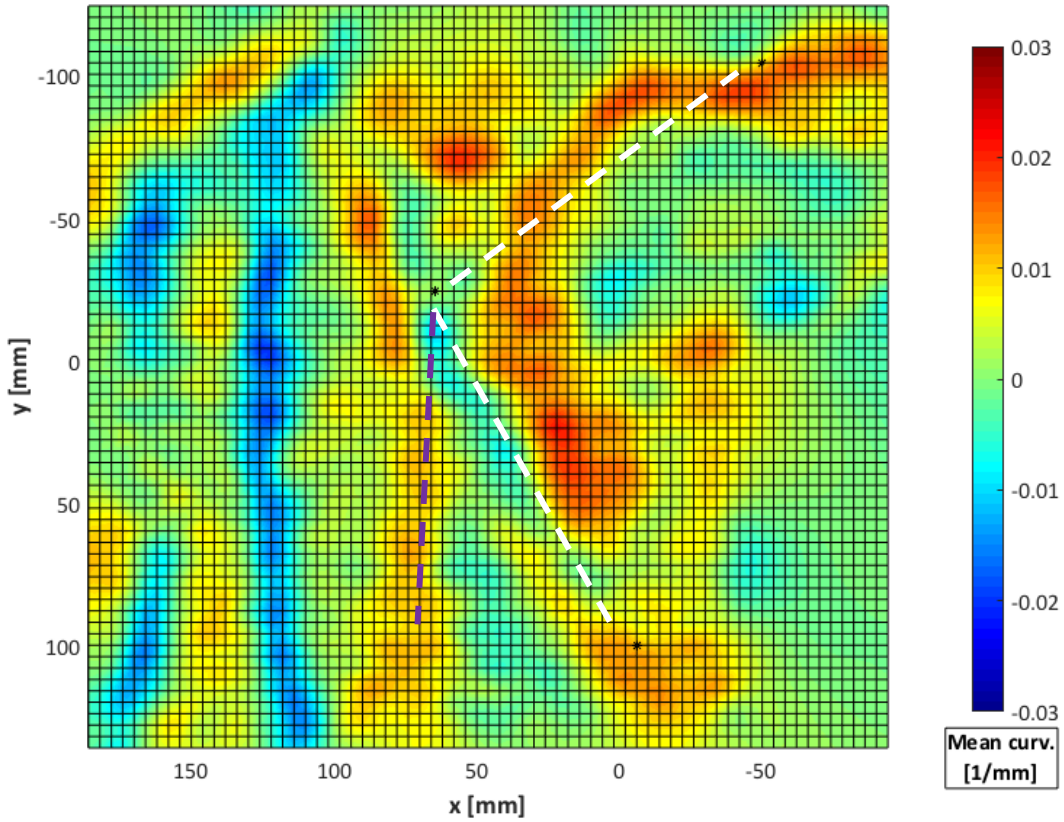


Figure 52: Participant 2 (M), 90° abduction (white line indicates scapula position at 90°, purple line indicates the position at 45°).

The first 30° of arm abduction are typically done without the involvement of the scapula, we can therefore expect the scapula to be rotated by approximately 15° on the SC map in Figure 51. The white and purple lines in the images are used to indicate the change in scapula position. Note again how the position of the scapula is correlated with the red areas of high positive curvature.

When the scapula starts to move with the arm, the surrounding musculature activates to move and stabilise the scapula. The activated muscles bodies bulge under the skin and cause the overall SC to increase along the entire medial border of the scapula and above the MB landmark.

In Figure 52 the scapula position at 45° is shown with a purple line to indicate the change in scapula position as the arm abducts from 45° to 90°. As with the other two cases, note how the SC heat map colours again change with the position of the scapula. The scapula position can be inferred from the curvature data. Notice how the IA becomes less visible as it moves into the lateral part of the body. In this position, the IA becomes difficult to palpate.

4.3.2 Less prominent scapula

An example of a participant (No. 4) with a scapula that does not have prominent borders is presented here. A less prominent scapula may represent a well-positioned scapula. Figure 53 shows the camera image with the arm at rest with the palpated IA, MB and AA marked on the skin. As in the example above, the SC data for the arm at rest is shown in Figure 54.

When Figure 54 is compared with the same image (Figure 50) from the prominent scapula example, it can be seen that the overall curvature intensity is much lower. However, even though not visible to the naked eye, the position of the scapula is still detected using the SC data. A vague outline of the scapula is identifiable, as well as the location of the spinal column.

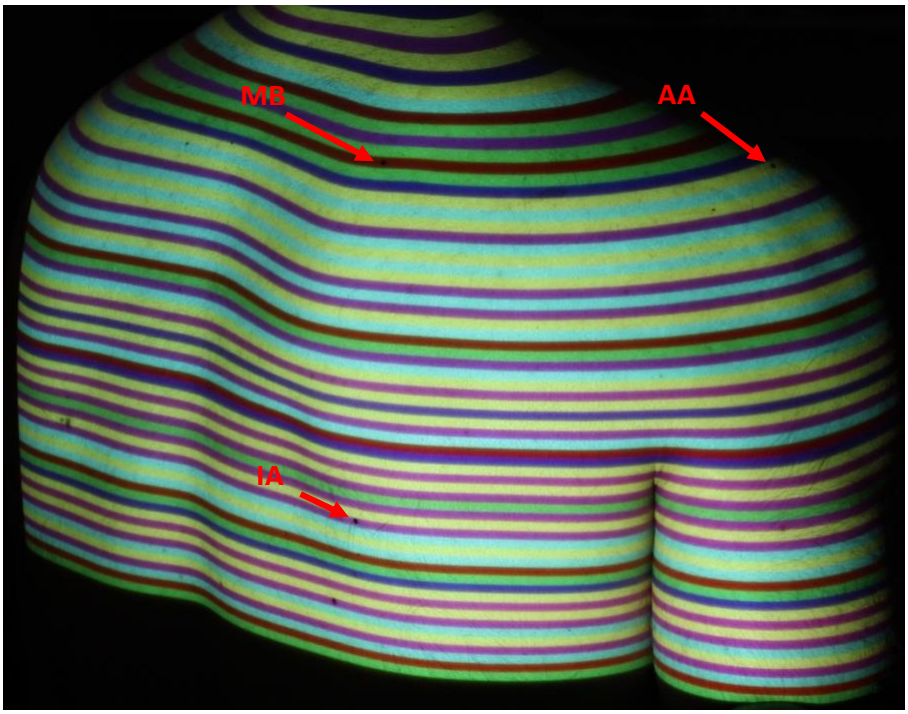


Figure 53: Participant 4 (M), camera image at 0° abduction.

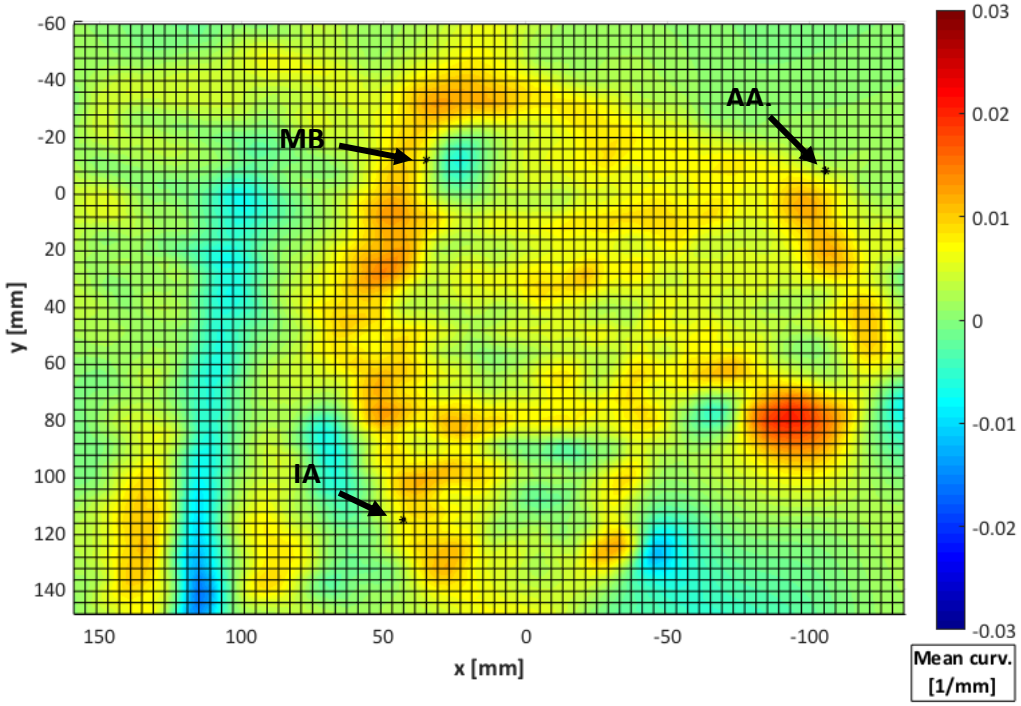


Figure 54: Participant 4 (M), SC at 0° abduction.

Figure 55 and Figure 56 show the SC maps for the less prominent scapula with the arm abducted to 45° and 90° respectively. Note how the curvature intensity increases next to the medial border as the muscles activate in both cases. In this example, the position of the scapula is not easily identifiable using only the curvature data. The small indentations around the MB and AA are the only clues to the location of these scapular landmarks. Note, however,

that the rotation of the scapula is clearly visible from the SC data, as shown by the dotted lines drawn between the locations of the palpated scapular landmarks in each image.

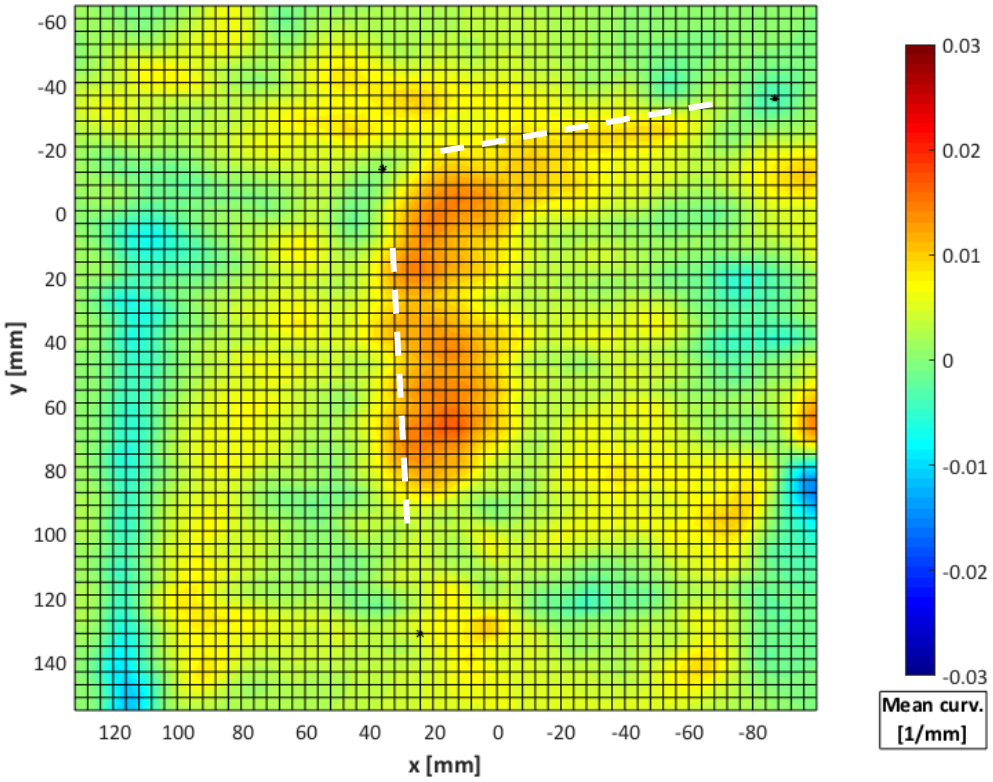


Figure 55: Participant 4 (M), SC at 45° abduction.

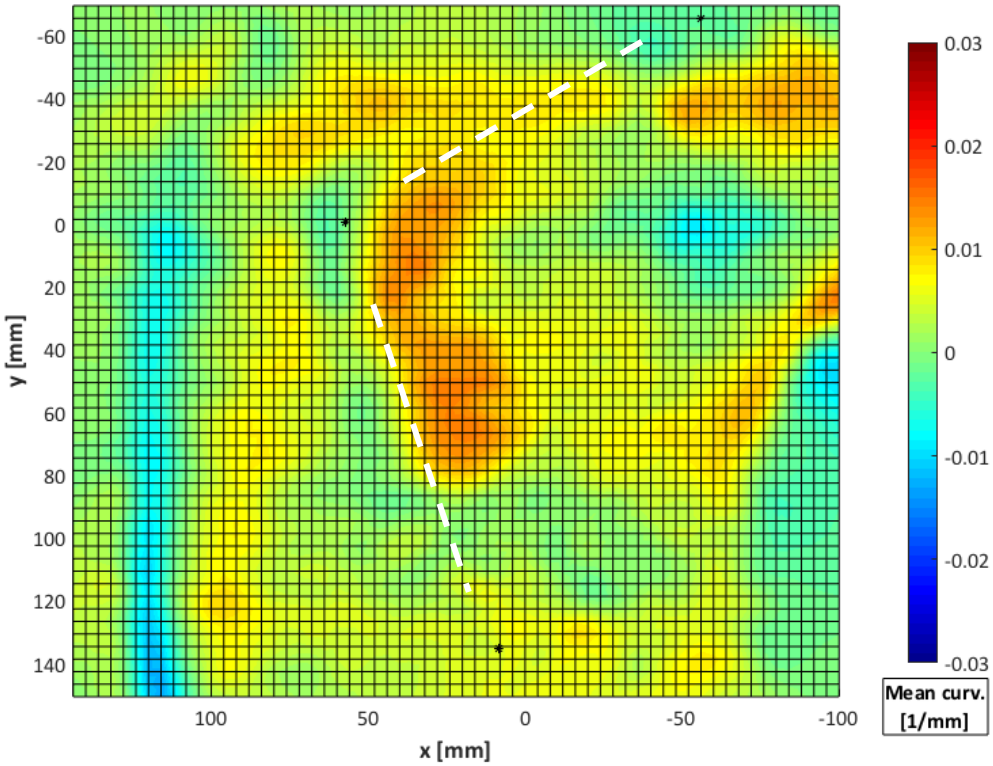


Figure 56: Participant 4 (M), SC at 90° abduction.

4.3.3 Muscular back

In this example, the participant (No. 7) had a high BMI but a H-W ratio that was well within limits. The participant had a very muscular back resulting in a thick layer of muscle tissue across the scapula. The participant had a less prominent scapula that was not easily identifiable by visual inspection. Figure 57 and Figure 58 show the camera image and SC data for the arm at rest.

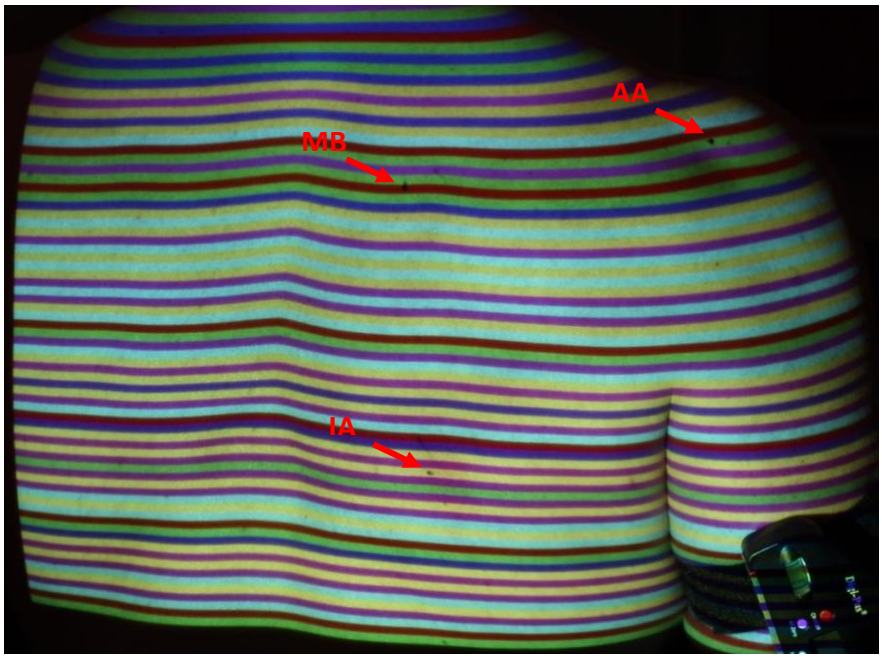


Figure 57: Participant 7 (M), camera image at 0° abduction.

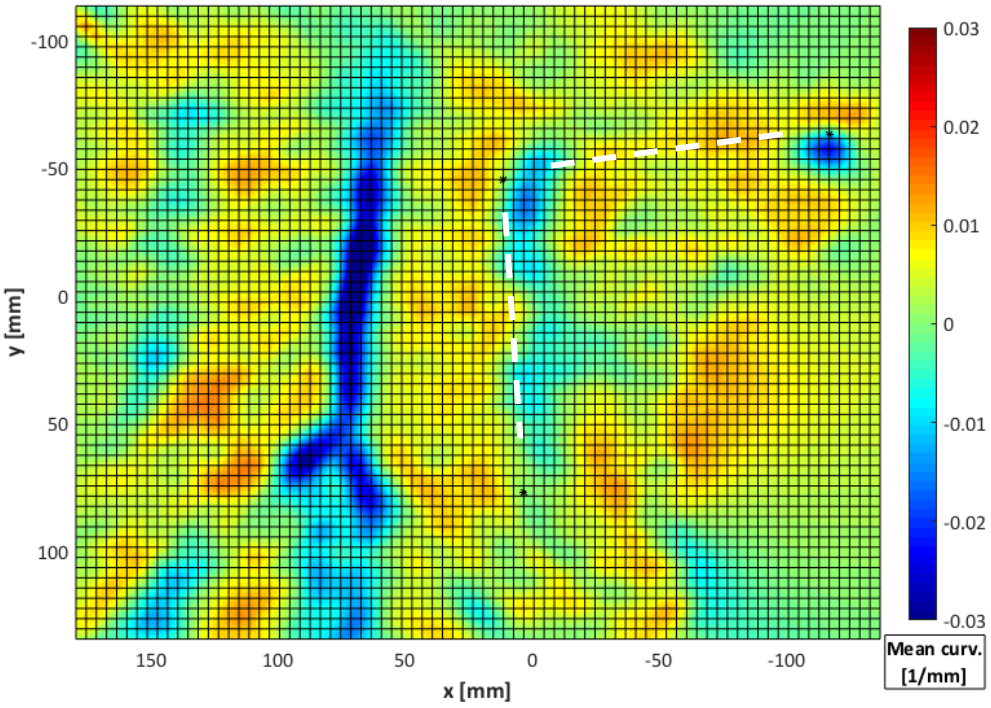


Figure 58: Participant 7 (M), SC at 0° abduction.

In this case the position of the scapula is visible due to the negative curvatures (valleys) instead of the peaks, as in the first case where the scapula was prominent. Figure 59 and Figure 60 show the curvature maps for the 45° and 90° positions.

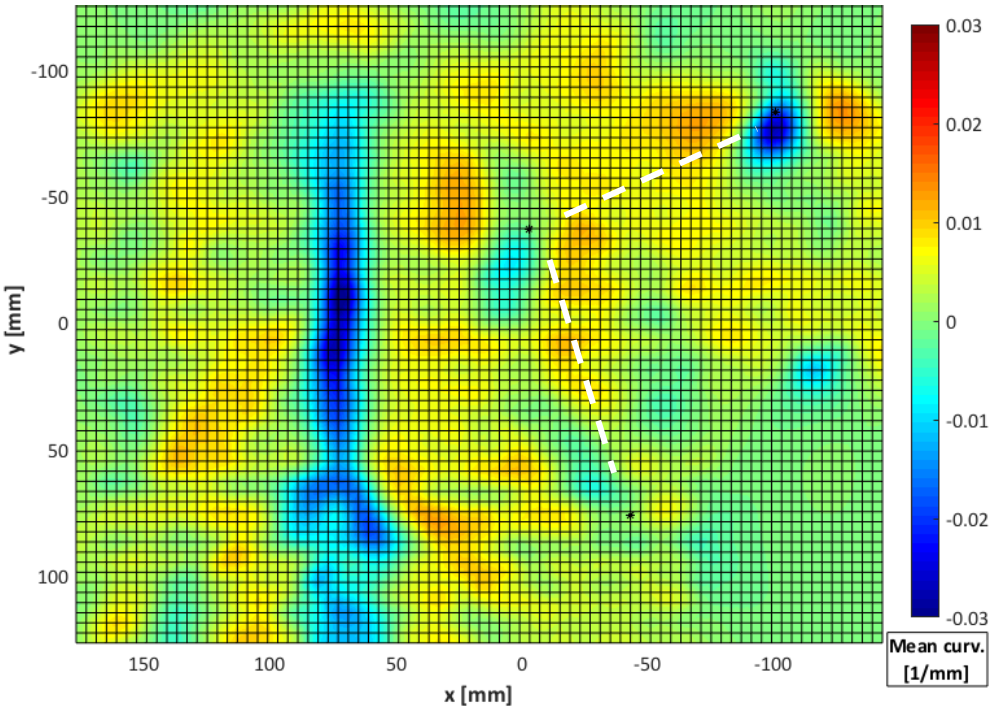


Figure 59: Participant 7 (M), SC at 45° abduction.

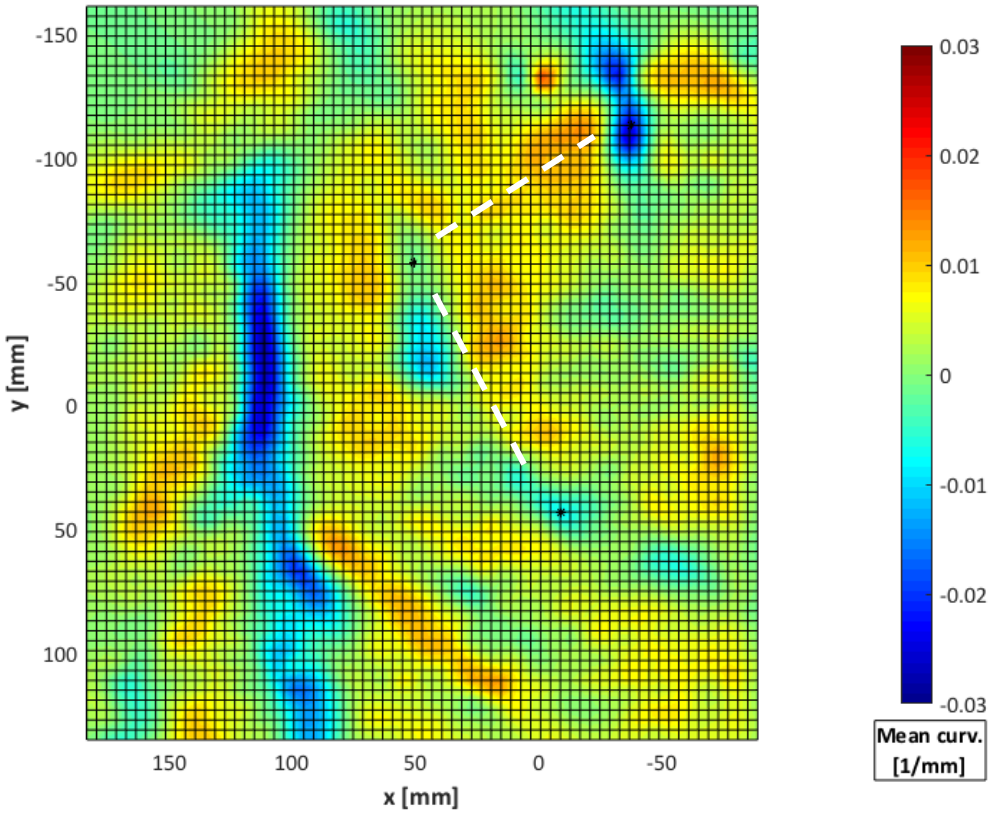


Figure 60: Participant 7 (M), SC at 90° abduction.

The muscle attach to the scapula in such a way that the inferior angle, medial border, spine and especially the AA is most exposed (see discussion in section 3.1.1). This creates valleys in the 3D topographic map that are highlighted by the SC data. In this case the AA was surrounded by a deep valley that stayed in more or less the same position throughout the arm movement range, this is a clear example of why the acromion marker cluster is currently the most accurate scapula tracker. Even though the position of the scapula was not visible to the naked eye, the SC could still be used to get a rough estimate of the scapula's location using the valleys created by the muscle attachments.

4.3.4 Thick subcutaneous fat layer

In this final case, a participant (No. 8) with a very high BMI is presented. A high BMI is usually associated with a thick layer of fat underneath the skin. This fat layer can obscure the scapula and greatly diminish the visibility of any protrusions. Figure 61 shows the camera image of the arm at rest with the palpated scapular landmarks and the SC data for the arm at rest is shown in Figure 62.

In this case, the scapular location could not be visually identified, firstly because the individual had a less prominent scapula, and secondly because of a relatively thick layer of sub-cutaneous fat that covered the scapular landmarks. With the arm at rest, the SC map showed a slight increased curvature near the IA, but the other landmarks could not be linked to any curvature features (Figure 62).

As the arm was moved to 45° abduction the underlying scapular muscles activated and the SC map showed additional features (Figure 63). Interestingly, it was only when the arm was moved to 90° abduction (Figure 64), and these features moved with the scapula, that it became clear which features could be linked to the position of the scapula. Similar to the case of the less prominent scapula (Figure 55 and Figure 56), the muscle activity to the right of the medial border created a zone of red peaks while a clear valley formed to the left of the medial border between the scapula and the spine. Even though the exact location of the scapula cannot be determined using the SC map, it provides a good estimate of the actual position of the scapula.

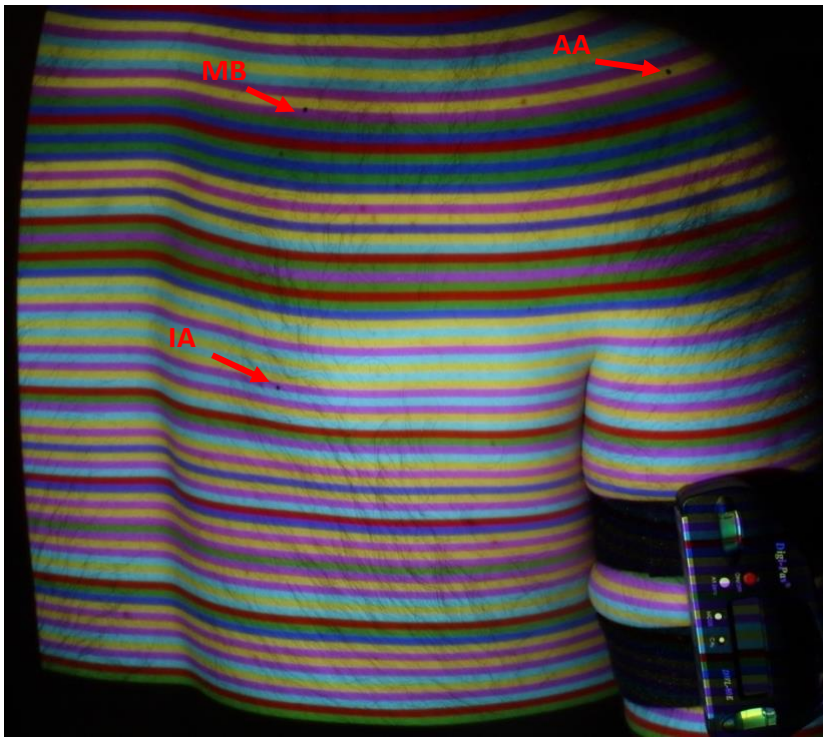


Figure 61: Participant 8 (M), camera image at 0° abduction.

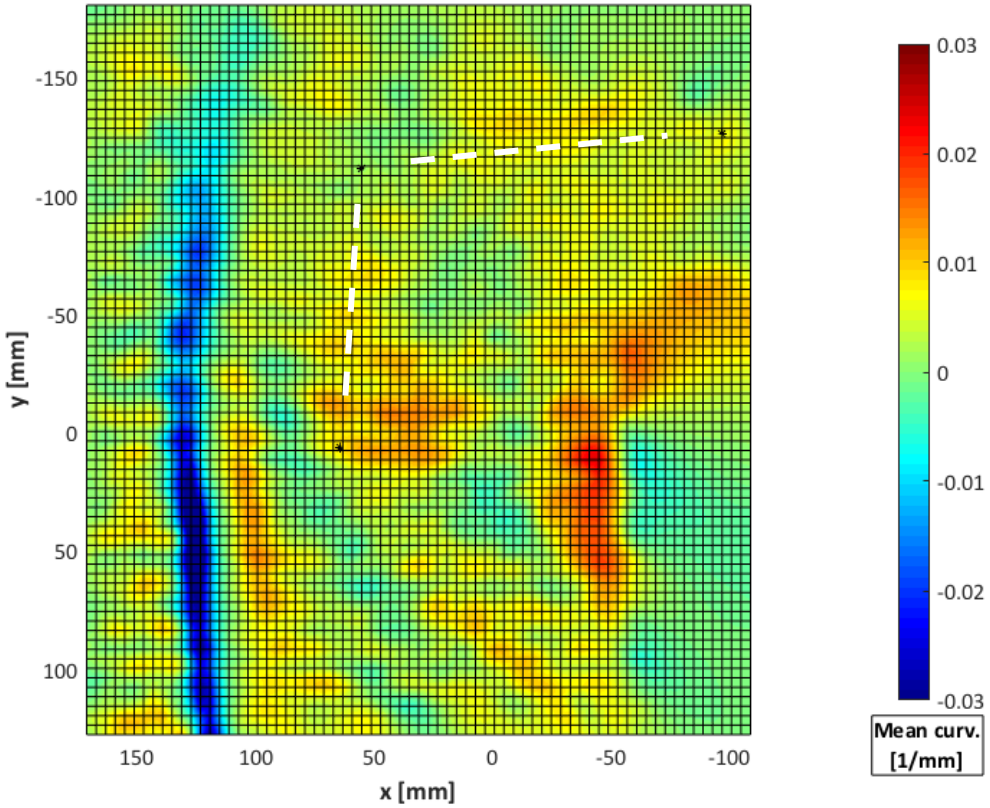


Figure 62: Participant 8 (M), SC at 0° abduction.

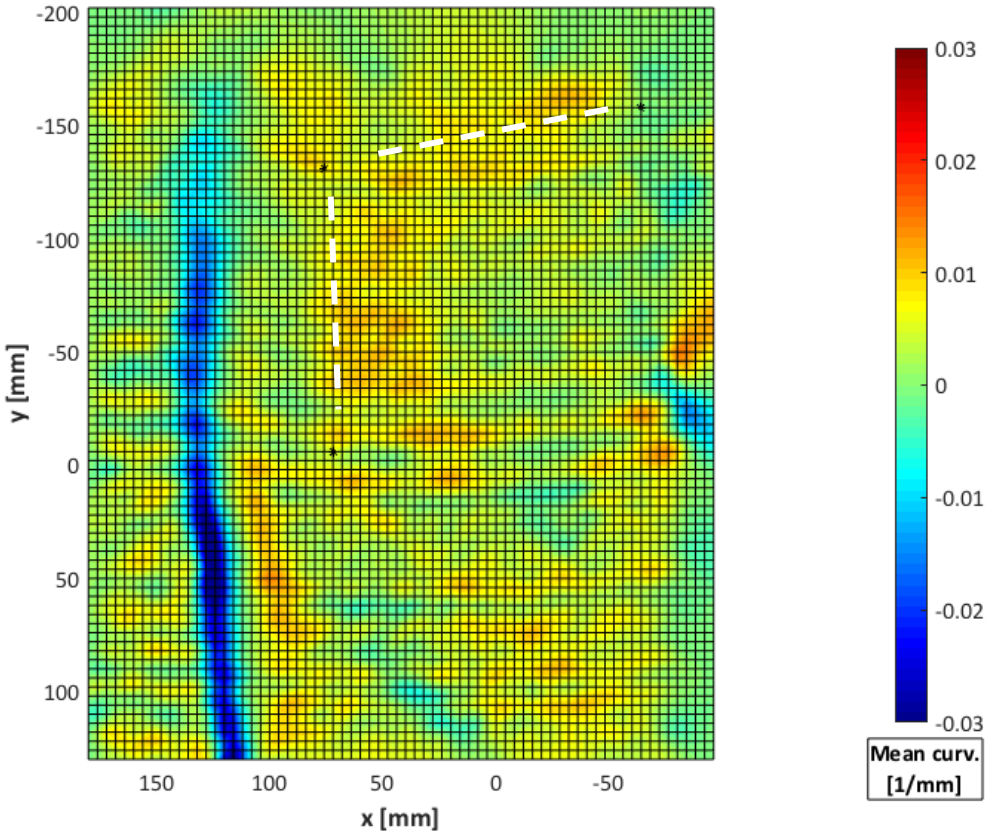


Figure 63: Participant 8 (M), SC at 45° abduction.

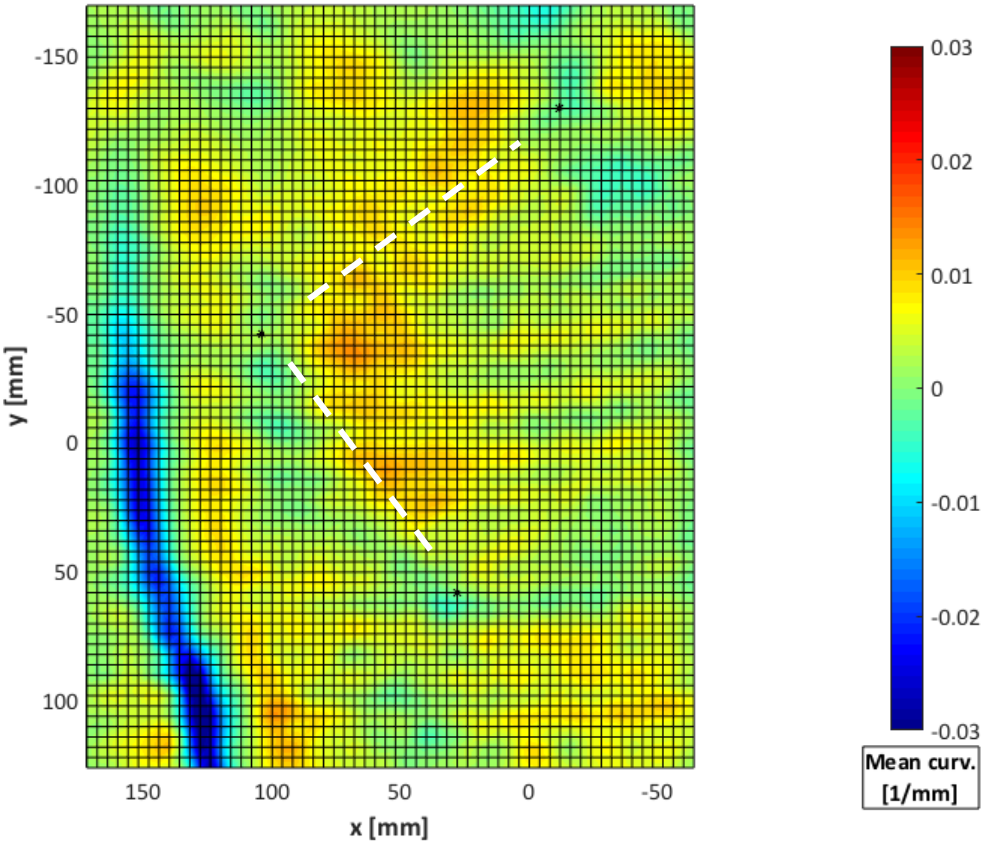


Figure 64: Participant 8 (M), SC at 90° abduction.

4.4 Clinical variables

4.4.1 Soft-tissue covering

A major issue that plagues all scapula tracking methods is STAs. STAs are created when a cutaneous marker is used to record the incorrect location of a bony landmark due to a movement deficit that exists when the bony structure moves more (or less) than the skin above it. The main issue with STAs is the fact that the size of the measurement error is unknown. It is always assumed that there will be some error but there is no way of measuring it non-invasively.

The SC approach effectively eliminates this difficulty because it does not measure a marker location with the assumption that the marker represents a specific landmark. The location of the underlying bony structure is estimated directly from the surface data that represents the skin position at a specific moment in time. When the underlying structure moves, the overlying soft tissue move in reaction to it and create a new surface map with unique SC data. This method is, however, not without its own limitations and drawbacks. For example, the exact location of the underlying bony landmarks can be difficult to estimate using only the SC data and may not even be possible in some cases. If the underlying bony structure does not protrude the skin enough to be detectable, then no estimation is possible. This fact limits the applications of the SC method to individuals without excessive soft-tissue over the scapula.

4.4.2 Skin reflectance

Another anticipated difficulty that goes with any optical surface mapping method is the effect of the scanned object's surface reflectance. In our case the object is the human body and more specifically human skin. This aspect could not be simulated in the laboratory (as discussed in section 3.4.3). Fortunately, the scanner proved to be effective on a variety of different skin tone and reflectance combinations. This is mostly thanks to the simplified colour-correction technique that was used.

The scanner was able to successfully scan and reconstruct all of the participants, including those with darker and light skin tones. The most challenging variable of this study proved to be the oiliness of the skin. Skin types with a high oil content created reflections that interfered

with the edges detection algorithm in some cases. This was, however, easily detected during the image capture phase and could be corrected by adjusting the position of the participant slightly.

4.5 Diagnostic potential

The main goal of this project was to develop a measurement tool for scapular dyskinesis and test it in a clinical environment. Current diagnostic methods involve either a 4-type classification, a Yes/No classification or the scapular dyskinesis test (SDT) (Rossi, Pedroni, Martins, & De Oliveira, 2017), which are primarily based on visual observations of the scapula. Observations are typically done at rest and during arm movements to detect any prominences or asymmetries and rely heavily on visual observations.

The main challenge in developing a tool that can measure scapular dyskinesis was in deciding what to measure. Because the current visual observations focus on scapula prominences and asymmetries, our proposed method attempted to quantify these prominences and asymmetries using a SC analysis technique. Prominences are not only highlighted and made more visible using the SC methods, but a measurement of the mean curvature is also done. Although this study was limited to the analysis of one side (right) only, it could easily be expanded to include both scapulae to detect and measure asymmetries. It is anticipated that our SC method would allow the detection of small prominences and asymmetries that are currently undetectable using only visual observations.

To further illustrate the potential diagnostic application of our SC method, an example of how the mean curvature can be used to quantify scapular protrusions is given. Firstly, consider the SC data for participant 2, with the arm at rest (Figure 65). In this image, the mean curvature values (H) are shown at each of the scapular landmarks as $H_{IA} = 0.017mm^{-1}$, $H_{MB} = 0.020mm^{-1}$ and $H_{AA} = 0.019mm^{-1}$ respectively.

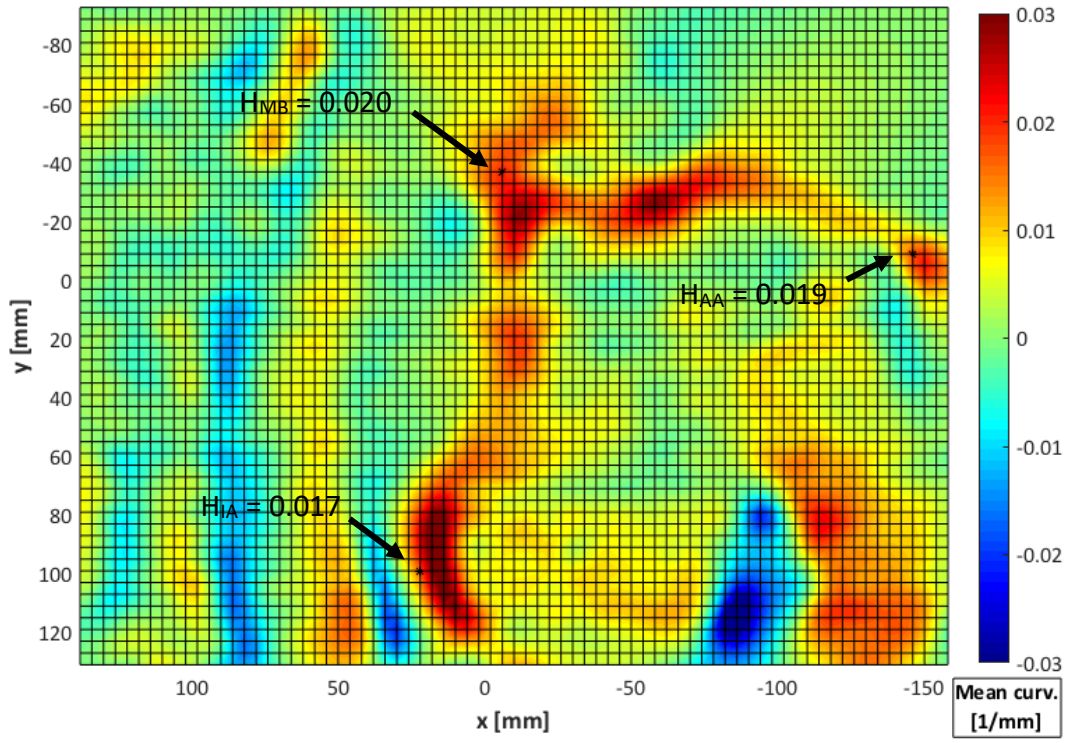


Figure 65: Participant 2 (M), 0° abduction.

The mean curvature values for the IA, MB and AA were recorded for all participants at three different arm positions (at rest, 45° and 90° abduction). The results for the arm at 0° abduction are shown here as it has been shown that scapular dyskinesis is most often detected in the resting position (Deng, Chen, Ma, Chen, & Huang, 2017). The results are shown in Table 9.

Table 9: Mean curvature values for all participants (arm at rest).

Participant	1	2	3	4	5	6	7	8	9	10	11	12
Gender	M	M	M	M	M	M	M	M	F	F	F	F
Prominent scapula?	yes	yes	no	no	no	no	no	no	yes	no	no	yes
H_{IA} [mm^{-1}]	0,028	0,017	0,014	0,007	0,007	0,013	0,000	0,009	0,026	0,012	0,008	0,021
H_{MB} [mm^{-1}]	0,008	0,020	0,012	0,006	0,007	0,002	0,002	0,002	0,015	0,017	0,001	0,019
H_{AA} [mm^{-1}]	0,005	0,019	0,007	0,004	0,005	0,002	-0,016	0,007	0,030	0,011	0,003	0,008

We now plot the mean curvature results, as measured at these scapular landmarks for all participants, with the arm at rest (Deng et al., 2017).

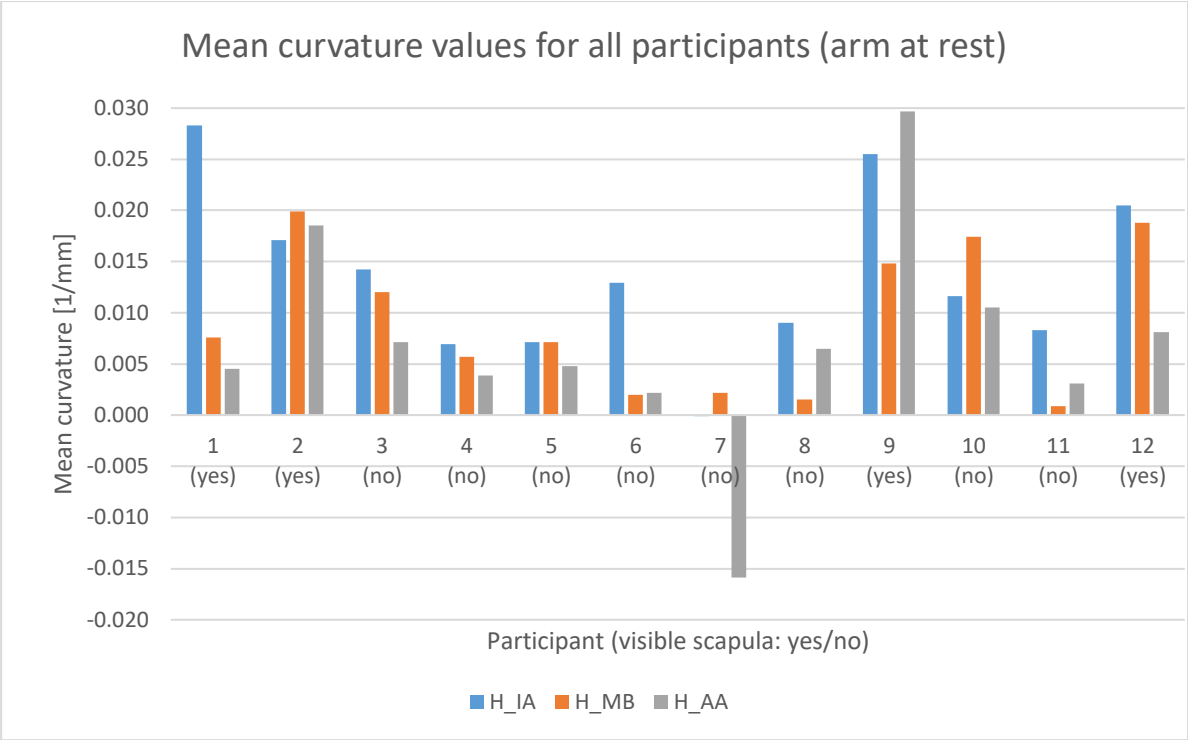


Figure 66: Mean curvature on the scapular landmark for all participants (arm at rest).

Note how all the participants with a clearly visible scapula at rest (participants 1, 2, 9 and 12) had at least one scapular landmark with a curvature value equal to or above 0.02mm^{-1} while the participants with less prominent or undetectable scapulae all exhibit curvature values below 0.015mm^{-1}. Also take note of the negative curvature value at the AA for participant 7 who presented with a very muscular back.

The above example is one proposed way of using the SC method to potentially screen patients as Yes/No for the prevalence of scapular dyskinesis. The landmark-specific SC data can then also be used to classify the dyskinesis according to Type (based on which parts of the scapula are prominent). In addition to a screening, the pre- and post-intervention SC data can also potentially be used to monitor a patient's progress and analyse the effectiveness of a treatment protocol.

Furthermore, our SC method can also be used to do surface motion analysis at 70 frames per second. Unfortunately, such an extensive analysis was outside the scope of this project, but a motion analysis would potentially further increase the diagnostic potential of the tool since a frame-to-frame change in surface topography would reveal even the smallest subcutaneous movements. With the data processing capabilities of this project, a six second video of a single

arm elevation and lowering would require a few hours of computational time to generate the mean curvature data for the whole video. A faster processor and further programming optimisation may bring this time down to a couple of minutes.

5 Conclusion and Recommendations

The main objective of this study was to develop a measurement tool for scapular dyskinesis. A low-cost 3D structured light scanner was developed using an off-the-shelf camera and projector. This scanner was employed to capture a 3D surface map of the upper back across the scapula and effectively record the skin protrusion caused by the underlying scapula. Furthermore, a surface analysis methodology was developed to analyse and measure the skin protrusions. Finally, it was shown how this tool could potentially be used in a clinical environment to screen and diagnose scapular dyskinesis. A full validation study will need to be conducted in order to confirm the final diagnostic potential of the tool.

5.1 Major conclusions drawn from the study

Some of the major conclusions drawn from this study include:

- A low-cost 3D structured light scanner can be built using a camera and projector and such a 3D scanner can effectively be used to scan and reconstruct the human body.
- The main challenge with using a structured light scanner on humans is related to the skin's reflectance properties. Fortunately, this can be overcome by employing colour correction strategies.
- The developed tool could be applied to a wide range of participants with varying BMIs and skin reflectance properties.
- The developed tool could be successfully used to identify scapular protrusions and measure the mean curvature at these points.
- This study provides a proof of concept for a tool that can be used to measure key clinical variables that can potential be used in future evaluations as well as for pre- and post-intervention comparisons.

Some limitations identified during this study include:

- Calculating the mean curvature data for a video at 70 frames per second would require excessive computational time – further optimisations are required to reduce this time and provide a practical solution. Programming in C+ or Python as opposed to Matlab may produce faster runtimes.
- The tripod used for the camera and projector need to be of a high quality to avoid small, slow movement in the system during and after calibration.
- A glass calibration board is very fragile and may easily break during transport. In addition to this, the glass surface can be prone to reflections which need to be avoided in order to ensure an accurate calibration.
- Marking on the skin is still required during image collection, true marker-less approach seems out of reach at this stage.

5.2 Future work and recommendations

Other aspects and future developments that fall outside the scope of this project include:

- This proof of concept needs to be taken further to be validated against a known tool for analysing and quantifying scapula position such as fluoroscopy or CT scan. This needs to be done to investigate and fully characterise the relationship between surface curvature data and scapular dyskinesis.
- To avoid repetitive calibrations, it is suggested that the chosen camera and projector be as small as possible and securely mounted on a rigid structure. This will enable a single calibration and a potential portable scanner setup.
- The applications of a 3D body surface scanner extend far beyond the use on the scapula only. The 3D surface mapping technology, in particular, can be extended to reconstruct any other body part. This has potential applications for foot analysis, facial reconstruction, scoliosis detection, etc.
- As a suggested next step, a CT/X-ray scanner can be used to measure the protrusions of the scapular landmarks, i.e. how far they lift off of the rib cage. These protrusion measurements can be then be linked to the SC measurements to quantify the mean curvature measurements in relation to protrusion distance.

- This scanner also needs to be tested in an injured population and against the visual observation tool to confirm the accuracy of the device in identifying individuals with dyskinesis.
- A 3D surface mapping approach can be used to analyse other prominent bony anatomy and support a statistical shape modelling pipeline

6 Reference list

Artec 3D. (2017). Artec Eva Lite. Retrieved October 27, 2017, from https://www.artec3d.com/3d-scanner/artec-eva-lite

Bayer, B. E. (1976). *US3971065*. United States.

Best Performance Group. (2014). Shoulder Complex – Kinematic Considerations [Image]. Retrieved August 15, 2016, from http://bestperformancegroup.com/?page_id=966

Bey, M. J., Zauel, R., Brock, S. K., & Tashman, S. (2006). Validation of a new model-based tracking technique for measuring three-dimensional, in vivo glenohumeral joint kinematics. *Journal of Biomechanical Engineering*, *128*(4), 604–609. https://doi.org/10.1115/1.2206199

Bouguet, J.-Y. (2015). Camera Calibration Toolbox for Matlab. Retrieved March 16, 2017, from http://www.vision.caltech.edu/bouguetj/calib_doc/

Brimijoin, W. O., & Geest, J. van der. (2011). de Bruijn sequence generator. Retrieved October 27, 2016, from https://www.mathworks.com/matlabcentral/fileexchange/28015-de-bruijn-sequence-generator?focused=5193893&tab=function

Brochard, S., Lempereur, M., & Rémy-Néris, O. (2011). Double calibration: An accurate, reliable and easy-to-use method for 3D scapular motion analysis. *Journal of Biomechanics*, *44*(4), 751–754. https://doi.org/10.1016/j.jbiomech.2010.11.017

Brooks, P. M. (2006). The burden of musculoskeletal disease—a global perspective. *Clinical Rheumatology*, *25*(6), 778–781. https://doi.org/10.1007/s10067-006-0240-3

Burkhart, S. S., Morgan, C. D., & Kibler, W. Ben. (2003). The disabled throwing shoulder: Spectrum of pathology part III: The SICK scapula, scapular dyskinesis, the kinetic chain, and rehabilitation. *Arthroscopy - Journal of Arthroscopic and Related Surgery*, *19*(6), 641–661. https://doi.org/10.1016/S0749-8063(03)00389-X

Carse, B., Meadows, B., Bowers, R., & Rowe, P. (2013). Affordable clinical gait analysis: An assessment of the marker tracking accuracy of a new low-cost optical 3D motion analysis system. *Physiotherapy (United Kingdom)*, *99*(4), 347–351. https://doi.org/10.1016/j.physio.2013.03.001

Caspi, D., Kiryati, N., & Shamir, J. (1998). Range Imaging with Adaptive Color Structured Light Range Imaging with Adaptive Color Structured Light. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *20*(5), 470–480.

Charbonnier, C., Chague, S., Kolo, F. C., Chow, J. C. K., & Ladermann, A. (2014). A patient-specific measurement technique to model shoulder joint kinematics. *Orthopaedics and Traumatology: Surgery and Research*, *100*(7), 715–719. https://doi.org/10.1016/j.otsr.2014.06.015

Claxton, D. (2006). Surfature Curvature. Retrieved July 18, 2017, from https://www.mathworks.com/matlabcentral/fileexchange/11168-surface-curvature

Deng, S., Chen, K., Ma, Y., Chen, J., & Huang, M. (2017). The Influence of Test Positions on Clinical Assessment for Scapular Dyskinesis. *PM and R*, *9*(8), 761–766. https://doi.org/10.1016/j.pmrj.2016.11.011

Drerup, B. (2014). Rasterstereographic measurement of scoliotic deformity. *Scoliosis*, *9*(1), 22. https://doi.org/10.1186/s13013-014-0022-7

Eiríksson, E. R., Wilm, J., Pedersen, D. B., & Aanæs, H. (2016). Precision and Accuracy Parameters in Structured Light 3-D Scanning, *XL*(December 2015), 1–2. https://doi.org/10.5194/isprsarchives-XL-5-W8-7-2016

G. Falcao, N. Hurtos, J. Massich, D. F. (2009). Projector-Camera Calibration Toolbox. Retrieved March 16, 2017, from http://code.google.com/p/procamcalib

Geng, J. (2011). Structured-light 3D surface imaging: a tutorial. *Advances in Optics and Photonics*, *3*, 128–160. https://doi.org/10.1364/AOP.3.000128

Gomes, P. F., Sesselmann, M., Faria, C. D. C. M., Araújo, P. A., & Teixeira-Salmela, L. F. (2010). Measurement of scapular kinematics with the moiré fringe projection technique. *Journal of Biomechanics*, *43*(6), 1215–1219. https://doi.org/10.1016/j.jbiomech.2009.12.015

Herakleous, K., & Poullis, C. (2014). 3DUNDERWORLD-SLS: An Open-Source Structured-Light Scanning

System for Rapid Geometry Acquisition, (September 2016).
iReviews. (2014a). 3D Digital Corp eScan3D. Retrieved October 27, 2017, from https://3d-scanners.www1.ireviews.com/3d-digital-corp-escan3d-review
iReviews. (2014b). Nikon ModelMaker MMCx. Retrieved October 27, 2017, from https://3d-scanners.www1.ireviews.com/nikon-modelmaker-mmcx-review
John Michael. (2008). Prosthetic Principles Upper Extremity Amputations [Image]. Retrieved August 14, 2016, from http://www.oandplibrary.org/reference/uclamanual/UCLA-02.pdf
Kibler, W. Ben. (1998). The Role of the Scapula in Athletic Shoulder Function. *The American Journal of Sports Medicine*, *26*(2), 325–337. https://doi.org/10.1177/0363546598026002801
Kibler, W. Ben, Ludewig, P. M., McClure, P. W., Michener, L. A., Bak, K., & Sciascia, A. D. (2013). Clinical implications of scapular dyskinesis in shoulder injury: the 2013 consensus statement from the "Scapular Summit". *British Journal of Sports Medicine*, *47*(14), 877–885. https://doi.org/10.1136/bjsports-2013-092425
Kibler, W. Ben, & McMullen, J. (2003). Scapular dyskinesis and its relation to shoulder pain. *Journal of the American Academy of Orthopaedic Surgeons*, *11*(2), 142–151. Retrieved from http://journals.lww.com/jaaos/Fulltext/2003/03000/Scapular_Dyskinesis_and_Its_Relation_to_Shoulder.8.aspx
Kroon, D.-J. (2010). Non-local Neans Noise Filter. Retrieved November 29, 2016, from https://www.mathworks.com/matlabcentral/fileexchange/27395-fast-non-local-means-1d--2d-color-and-3d
Lanman, D. (2009a). Ray-plane intersection [Image]. Retrieved August 10, 2017, from https://www.slideshare.net/dlanman/build-your-own-3d-scanner-the-mathematics-of-3d-triangulation-1882923
Lanman, D. (2009b). Structured Light for 3D Scanning. Retrieved November 16, 2016, from http://mesh.brown.edu/byo3d/source.html
Lempereur, M., Brochard, S., Leboeuf, F., & Rémy-Néris, O. (2014). Validity and reliability of 3D marker based scapular motion analysis: A systematic review. *Journal of Biomechanics*, *47*(10), 2219–2230. https://doi.org/10.1016/j.jbiomech.2014.04.028
Lewis, J., Green, A., Reichard, Z., & Wright, C. (2002). Scapular position: The validity of skin surface palpation. *Manual Therapy*, *7*(1), 26–30. https://doi.org/10.1054/math.2001.0405
Li, H. (2004). *Reconstruction Using Structured Light*.
LoveOfDrawing. (2016). Scapula [Image]. Retrieved August 31, 2016, from http://www.loveofdrawing.com/images/Scapula.jpg
Matsui, K., Shimada, K., & Andrew, P. D. (2006). Deviation of skin marker from bone target during movement of the scapula. *Journal of Orthopaedic Science*, *11*(2), 180–184. https://doi.org/10.1007/s00776-005-1000-y
Mattson, J. M., Russo, S. A., Rose, W. C., Rowley, K. M., & Richards, J. G. (2012). Identification of scapular kinematics using surface mapping: A validation study. *Journal of Biomechanics*, *45*(12), 2176–2179. https://doi.org/10.1016/j.jbiomech.2012.05.048
McClure, P., Michener, L. A., Sennett, B. J., & Karduna, A. R. (2001). Direct 3-dimensional measurement of scapular kinematics during dynamic movements in vivo. *Journal of Shoulder and Elbow Surgery*, *10*(3), 269–277. https://doi.org/10.1067/mse.2001.112954
McClure, P., Tate, A. R., Kareha, S., Irwin, D., & Zlupko, E. (2009). A clinical method for identifying scapular dyskinesis, part 1: Reliability. *Journal of Athletic Training*, *44*(2), 160–164. https://doi.org/10.4085/1062-6050-44.2.160
Metzger, J. (2013). Protruding scapula [Image]. Retrieved February 17, 2018, from http://agingathletesblog.blogspot.co.za/2013/07/cycling-posturetrying-to-wrap-this-ting.html
Mutsvangwa, T., Burdin, V., Schwartz, C., & Roux, C. (2015). An Automated Statistical Shape Model Developmental Pipeline: Application to the Human Scapula and Humerus. *IEEE Transactions on Biomedical Engineering*, *62*(4), 1098–1107. https://doi.org/10.1109/TBME.2014.2368362
OpenCV. (2017). Computer Vision Library (C++, Python, Java). Retrieved from http://opencv.org
OpenStax. (2013). Figure of Scapula [Image]. Retrieved from

https://opentextbc.ca/anatomyandphysiology/
Optitrack. (2017). Optitrack motion capture systems. Retrieved October 27, 2017, from http://optitrack.com
Oster, G. (1988). *Optics source book - science reference series*. (S. Parker, Ed.). McGraw-Hill, USA.
Pagès, J., Salvi, J., Collewet, C., & Forest, J. (2005). Optimised de Bruijn patterns for one-shot shape acquisition. *Image and Vision Computing*, *23*(8), 707–720. https://doi.org/10.1016/j.imavis.2005.05.007
Paine, R., & Voight, M. L. (2013). The role of the scapula. *International Journal of Sports Physical Therapy*, *8*(5), 617–629. https://doi.org/10.2519/jospt.1993.18.1.386
Park, J.-Y. Y., Hwang, J.-T. T., Kim, K.-M. M., Makkar, D., Moon, S. G., & Han, K.-J. J. (2013). How to assess scapular dyskinesis precisely: 3-dimensional wing computer tomography-a new diagnostic modality. *Journal of Shoulder and Elbow Surgery*, *22*(8), 1084–1091. https://doi.org/10.1016/j.jse.2012.10.046
PFG Building Glass. (2017). Float glass manufacturing. Retrieved September 3, 2017, from http://pfg.co.za/product/float-glass/
Porto, F. F., Gurgel, J. L. L., Russomano, T., & Farinatti, P. D. T. V. (2010). Moiré topography: Characteristics and clinical application. *Gait & Posture*, *32*(3), 422–424. https://doi.org/10.1016/j.gaitpost.2010.06.017
Prinold, J. A. I., Shaheen, A. F., & Bull, A. M. J. (2011). Skin-fixed scapula trackers: A comparison of two dynamic methods across a range of calibration positions. *Journal of Biomechanics*, *44*(10), 2004–2007. https://doi.org/10.1016/j.jbiomech.2011.05.010
Rajamani, K. T., Styner, M. A., Talib, H., Zheng, G., Nolte, L. P., & Ballester, M. A. G. (2007). Statistical deformable bone models for robust 3D surface extrapolation from sparse data. *Medical Image Analysis*, *11*(2), 99–109. https://doi.org/10.1016/j.media.2006.05.001
Rossi, D. M., Pedroni, C. R., Martins, J., & De Oliveira, A. S. (2017). Intrarater and interrater reliability of three classifications for scapular dyskinesis in athletes. *PLoS ONE*, *12*(7), 1–10. https://doi.org/10.1371/journal.pone.0181518
Salvi, J., Pagès, J., & Batlle, J. (2004). Pattern codification strategies in structured light systems. *Pattern Recognition*, *37*(4), 827–849. https://doi.org/10.1016/j.patcog.2003.10.002
Seth, A., Matias, R., Veloso, A. P., & Delp, S. L. (2015). A biomechanical model of the scapulothoracic joint to accurately capture scapular kinematics during shoulder movements, 1–19. https://doi.org/10.1371/journal.pone.0141028
Studyblue. (2013). Muscles that move the prctoral girdle [Imange]. Retrieved July 25, 2017, from https://www.studyblue.com/notes/note/n/intro-and-muscles-that-move-the-pectoral-girdle-and-upper-limbs/deck/3165475
Thabet, A. K., Trucco, E., Salvi, J., Wang, W., & Abboud, R. J. (2014). Dynamic 3D shape of the plantar surface of the foot using coded structured light: a technical report. *Journal of Foot and Ankle Research*, *7*(1), 5. https://doi.org/10.1186/1757-1146-7-5
Uhl, T. L., Kibler, W. B., Gecewich, B., & Tripp, B. L. (2009). Evaluation of clinical assessment methods for scapular dyskinesis. *Arthroscopy : The Journal of Arthroscopic & Related Surgery : Official Publication of the Arthroscopy Association of North America and the International Arthroscopy Association*, *25*(11), 1240–1248. https://doi.org/10.1016/j.arthro.2009.06.007
Van Kessel, J. (2013). *Shape from colored structured light and polarization [Master's thesis]*. Utrecht University. Retrieved from https://dspace.library.uu.nl/handle/1874/287113
Warner, M. B., Chappell, P. H., & Stokes, M. J. (2012). Measuring scapular kinematics during arm lowering using the acromion marker cluster. *Human Movement Science*, *31*(2), 386–396. https://doi.org/10.1016/j.humov.2011.07.004
Weisstein, E. (2015). Mean Curvature. Retrieved October 24, 2017, from http://mathworld.wolfram.com/MeanCurvature.html
Wikipedia. (2006). Bayer filter [Image]. Retrieved August 18, 2017, from https://en.wikipedia.org/wiki/Bayer_filter
Wikipedia. (2007). Shoulder girdle [Image]. Retrieved November 28, 2017, from

https://en.wikipedia.org/wiki/Shoulder_girdle
World Health Organization. (2008). *Waist Circumference and Waist-Hip Ratio: Report of a WHO Expert Consultation*. *WHO Press*. https://doi.org/10.1038/ejcn.2009.139
Zhang, L., Curless, B., & Seitz, S. M. (2002). Rapid Shape Acquisition Using Color Structured Light and Multi-Pass Dynamic Programming. In *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*. Retrieved from http://grail.cs.washington.edu/projects/moscan/
Zhang, S., & Huang, P. S. (2006). Novel method for structured light system calibration. *Optical Engineering*, *45*(8), 83601-1–8. https://doi.org/10.1117/1.2336196

7 Appendices

7.1 Appendix I: Matlab code

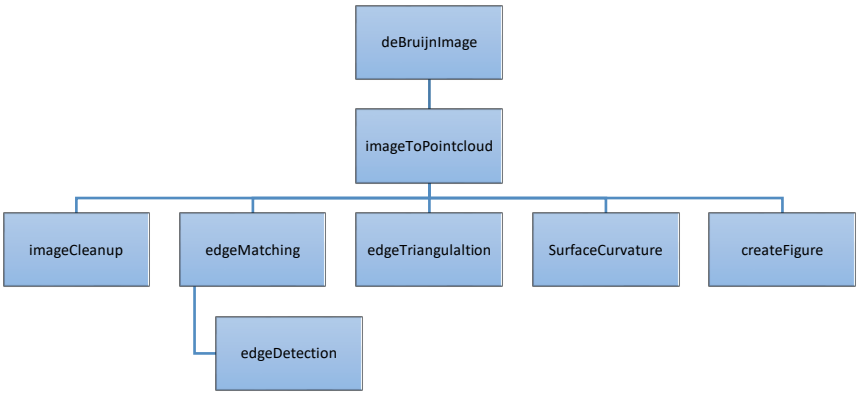


Figure 67: Matlab function layout.

```
function deB_I = deBruijnImage(col,row,colPixels,savefile)
%This function creates a col x row de Bruijn image  using 6 colours
%and a specified pixel-width for each colour band. A deBruijn pattern
%with k = 3 (the effective number of colours is 3 because of the XOR == 2
%requirement which means that each colour can effectively only be followed
%by two other colours. n = 5 is used to give enough stripes.
%A maximum of 243 (3^5) colour stripes can be used
%
%Variables:
%col/row = number of columns/rows
%colPixels = pixel width of each colour stripe
%Savefile is binary 1(yes) or 0(no)
%
%Created by: Jaco Verster (versterrie@gmail.com)

%Check stripe limit
val = ceil(col/colPixels);
if val >= 244
    disp(['ERROR: number of requested stripes (col/colPixels) is '...
        'too high! Lower col or increase colPixels'])
end

%specify resolution
%col = 70;
%row = 20;
%colPixels = 13;

%create memory for the image
deB_I = zeros(row,col,3);

%Create colour array using 6 colours (black and white was removed)
colM =  [1 1 0;  %yellow
         1 0 1;  %magenta
         0 1 1;  %cyan
         1 0 0;  %red
         0 1 0;  %green
         0 0 1];  %blue

%XOR operation for the specified deBruijn pattern.
debruijn = [0,0,0,0,0,1,0,0,0,0,2,0,0,0,1,1,0,0,0,1,2,0,0,0,2,1,0,0,0,...
    2,2,0,0,1,0,1,0,0,1,0,2,0,0,1,1,1,0,0,1,1,2,0,0,1,2,1,0,0,1,2,2,0,...
    0,2,0,1,0,0,2,0,2,0,0,2,1,1,0,0,2,1,2,0,0,2,2,1,0,0,2,2,2,0,1,0,1,...
    1,0,1,0,1,2,0,1,0,2,1,0,1,0,2,2,0,1,1,0,2,0,1,1,1,1,0,1,1,1,2,0,1,...
    1,2,1,0,1,1,2,2,0,1,2,0,2,0,1,2,1,1,0,1,2,1,2,0,1,2,2,1,0,1,2,2,2,...
    0,2,0,2,1,0,2,0,2,2,0,2,1,1,1,0,2,1,1,2,0,2,1,2,1,0,2,1,2,2,0,2,2,...
    1,1,0,2,2,1,2,0,2,2,2,1,0,2,2,2,2,1,1,1,1,1,2,1,1,1,2,2,1,1,2,1,2,...
    1,1,2,2,2,1,2,1,2,2,1,2,2,2,2,2];
```

```
pattern = zeros(1,length(debruijn));
pattern(1) = 1;
for i = 2:length(debruijn)
    if pattern(i-1) == 0
        if debruijn(i) == 0
            pattern(i) = 1;
        elseif debruijn(i) == 1
            pattern(i) = 2;
        elseif debruijn(i) == 2
            pattern(i) = 3;
        end

    elseif pattern(i-1) == 1
        if debruijn(i) == 0
            pattern(i) = 2;
        elseif debruijn(i) == 1
            pattern(i) = 0;
        elseif debruijn(i) == 2
            pattern(i) = 4;
        end

    elseif pattern(i-1) == 2
        if debruijn(i) == 0
            pattern(i) = 5;
        elseif debruijn(i) == 1
            pattern(i) = 1;
        elseif debruijn(i) == 2
            pattern(i) = 0;
        end

    elseif pattern(i-1) == 3
        if debruijn(i) == 0
            pattern(i) = 4;
        elseif debruijn(i) == 1
            pattern(i) = 5;
        elseif debruijn(i) == 2
            pattern(i) = 0;
        end

    elseif pattern(i-1) == 4
        if debruijn(i) == 0
            pattern(i) = 3;
        elseif debruijn(i) == 1
            pattern(i) = 1;
        elseif debruijn(i) == 2
            pattern(i) = 5;
        end

    elseif pattern(i-1) == 5
        if debruijn(i) == 0
            pattern(i) = 4;
        elseif debruijn(i) == 1
            pattern(i) = 3;
        elseif debruijn(i) == 2
            pattern(i) = 2;
        end
    end
end

%Fill in the colours
stripeColors = zeros(length(debruijn),3);
for i = 1:length(debruijn)
    if pattern(i) == 0
        stripeColors(i,1) = 0;
        stripeColors(i,2) = 0;
        stripeColors(i,3) = 1;

    elseif pattern(i) == 1
        stripeColors(i,1) = 0;
        stripeColors(i,2) = 1;
        stripeColors(i,3) = 0;

    elseif pattern(i) == 2
        stripeColors(i,1) = 1;
        stripeColors(i,2) = 0;
        stripeColors(i,3) = 0;
```

```
    elseif pattern(i) == 3
        stripeColors(i,1) = 1;
        stripeColors(i,2) = 1;
        stripeColors(i,3) = 0;

    elseif pattern(i) == 4
        stripeColors(i,1) = 1;
        stripeColors(i,2) = 0;
        stripeColors(i,3) = 1;

    elseif pattern(i) == 5
        stripeColors(i,1) = 0;
        stripeColors(i,2) = 1;
        stripeColors(i,3) = 1;
    end
end

%Create the image values
z_step = colPixels - 1;
for x = 1:row
    for y = colPixels:colPixels:col
        for z = -(z_step:-1:0)
            deB_I(x,y+z,1) = stripeColors(y/colPixels,1)*255;
            deB_I(x,y+z,2) = stripeColors(y/colPixels,2)*255;
            deB_I(x,y+z,3) = stripeColors(y/colPixels,3)*255;
        end
    end
end

figure, imshow(deB_I)

if ~exist('savefile','var')
    % third parameter does not exist, so default it to something
    savefile = 0;
end

if savefile == 1
    %save the image
    filename = ['debruijn_image_',num2str(row),'x',num2str(col),'x',...
        num2str(colPixels),'.png'];
    imwrite(deB_I, filename)
else
    %do nothing
end
```

```
%imageToPointcloud
%
%This script calls all the required functions that convert a camera image
%to a pointcloud. The pointcloud post-processing is also done using this
%script.

clc, clear ,close all

tic %timer started

imagename = '2017-11-15_mia_0b.JPG';
orientation = 'P'; %L = landscape, P = portrait

imageCleanup(imagename,1);

edgeMatching(imagename,orientation,1);

edgeTriangulation(imagename,1);

toc %timer stopped

pause;

%% Pointcloud post processing:
pointcloud = pcread(['pointCloud_',imagename,'clean.ply']);
%pcshow(pointcloud)
[gx,gy,gz,H,Pmax] = surfaceCurvature(imagename,pointcloud,0);

load(['matched_edges_(',imagename,').mat'])
```

```
load(['cleaned_up_image_(',imagename,').mat'])

pixels = selectPixels(Icrop,3)
xyz = pixelToXYZ(pixels,matched_cam_edges,vertices)

%This alternative "xyz" can be used when pixels are already pre-selected or
%saved previously:
    %points = stlread(['pointsOnSurface_',imagename,'.stl'])
    %xyz = points.vertices

surface = stlread(['pointSurface_',imagename,'.stl']);

%calculate distance from points to tri-mesh using point2trimesh.m
[distances,surface_points] = point2trimesh(surface, 'QueryPoints', xyz);
%point2trimesh function available from: https://www.mathworks.com/...
%.../matlabcentral/fileexchange/...
%.../52882-point2trimesh------distance%C2%A0between-point-and-triangulated-surface

%Extract Pmax location
[Pmax_max,I] = max(Pmax(:))
%Pmax_min = min(Pmax(:)) used for setting caxis values
[row_max,col_max] = ind2sub(size(Pmax),I)
[X,Y] = meshgrid(gx,gy);
XZslice = [gx; -gz(row_max,:)];
Pmax_point = [X(row_max,col_max) Y(row_max,col_max) gz(row_max,col_max)]

createfigure1(-gx, gy, -gz, H,...
    -surface_points(:,1), surface_points(:,2), -surface_points(:,3));
    %-xyz(:,1), xyz(:,2), -xyz(:,3),...
    %-Pmax_point(:,1), Pmax_point(:,2), -Pmax_point(:,3)

    %set plot axis limits
    xlim([min(-gx) max(-gx)])
    ylim([min(gy) max(gy)])

point_2_pointcloud_avg = mean(abs(distances))

stlwrite(['pointsOnSurface_',imagename,'.stl'],...
    surface_points(:,1),surface_points(:,2),surface_points(:,3))

%Distance from Pmax point to inferior angle mark
Dist = sqrt(sum((xyz(1,:)-Pmax_point).^2))

IA_col = find(gx <= (surface_points(1,1)+2) & gx > (surface_points(1,1)-2));
IA_row = find(gy <= (surface_points(1,2)+2) & gy > (surface_points(1,2)-2));
MB_col = find(gx <= (surface_points(2,1)+2) & gx > (surface_points(2,1)-2));
MB_row = find(gy <= (surface_points(2,2)+2) & gy > (surface_points(2,2)-2));
AA_col = find(gx <= (surface_points(3,1)+2) & gx > (surface_points(3,1)-2));
AA_row = find(gy <= (surface_points(3,2)+2) & gy > (surface_points(3,2)-2));

%plot x-z plane slice and y-z plane slice through Pmax
% figure, plot(gx,-gz(IA_row,:))
% title 'x-z plane slice through IA'
% xlabel('x'), ylabel('z')
% axis equal

% figure, plot(gx,-gz(MB_row,:))
% title 'x-z plane slice through MB'
% xlabel('x'), ylabel('z')
% axis equal

% figure, plot(gy,-gz(:,col_max))
% title 'y-z plane slice through Pmax'
% xlabel('y'), ylabel('z')
% axis equal
%}

%Show H value at IA, MB and AA
H_landmarks = [H(IA_row,IA_col); H(MB_row,MB_col); H(AA_row,AA_col)]
```

```
function imageCleanup(imagename, savefile)
%This function applies colour correction and non-linear means filtering to
%the input image is defined as "imagename" and savefile is either on (1) or
%off (0).
```

```
%
%Created by: Jaco Verster (versterrie@gmail.com)

%Manual override
%imagename = 'capture1.JPG';
%savefile = 0;

%Load the image into memory
I = im2double(imread(imagename));

disp('+++ imageCleanup initialized...')

%% Select bounding box
disp('+ Select bounding box...')

bounds = [];
figure, imagesc(I);
axis image off;
title('Select the bounding box'); drawnow;
[x,y] = ginput(2);
bounds = round([min(x) max(x) min(y) max(y)]);
hold on;
plot(bounds([1 2 2 1 1]),bounds([3 3 4 4 3]),'r.-');
hold off;
pause(1);
close

%Save cropped image to new variable
A = imcrop(I, ...
    [bounds(1) bounds(3) (bounds(2)-bounds(1)) (bounds(4)-bounds(3))]);

%% Colour correction
disp('+ Applying colour correction...')

%%%RED calibration
%select a image section with red vivble
boundsR = [];
figure, imagesc(A);
axis image off;
title('Crop to image section with RED clearly visible'); drawnow;
[x,y] = ginput(2);
boundsR = round([min(x) max(x) min(y) max(y)]);
hold on;
plot(boundsR([1 2 2 1 1]),boundsR([3 3 4 4 3]),'r.-');
hold off;
pause(1);
close

%Save cropped image to new variable
R_zoom = imcrop(A, ...
    [boundsR(1) boundsR(3) (boundsR(2)-boundsR(1)) (boundsR(4)-boundsR(3))]);

%select a clear red section
boundsR = [];
figure, imagesc(R_zoom);
axis image off;
title('Select a section representative of RED '); drawnow;
[x,y] = ginput(2);
boundsR = round([min(x) max(x) min(y) max(y)]);
hold on;
plot(boundsR([1 2 2 1 1]),boundsR([3 3 4 4 3]),'r.-');
hold off;
pause(1);
close

%Save cropped image to new variable
C{1} = imcrop(R_zoom, ...
    [boundsR(1) boundsR(3) (boundsR(2)-boundsR(1)) (boundsR(4)-boundsR(3))]);

%%%GREEN calibration image
%select a image section with green vivble
boundsG = [];
figure, imagesc(A);
axis image off;
title('Crop to image section with GREEN clearly visible'); drawnow;
[x,y] = ginput(2);
boundsG = round([min(x) max(x) min(y) max(y)]);
```

```
hold on;
plot(boundsG([1 2 2 1 1]),boundsG([3 3 4 4 3]),'r.-');
hold off;
pause(1);
close

%Save cropped image to new variable
G_zoom = imcrop(A, ...
    [boundsG(1) boundsG(3) (boundsG(2)-boundsG(1)) (boundsG(4)-boundsG(3))]);

%select a clear green section
boundsG = [];
figure, imagesc(G_zoom);
axis image off;
title('Select a section representative of GREEN'); drawnow;
[x,y] = ginput(2);
boundsG = round([min(x) max(x) min(y) max(y)]);
hold on;
plot(boundsG([1 2 2 1 1]),boundsG([3 3 4 4 3]),'r.-');
hold off;
pause(1);
close

%Save cropped image to new variable
C{2} = imcrop(G_zoom, ...
    [boundsG(1) boundsG(3) (boundsG(2)-boundsG(1)) (boundsG(4)-boundsG(3))]);

%%%BLUE calibration
%select a image section with blue vivble
boundsB = [];
figure, imagesc(A);
axis image off;
title('Crop to image section with BLUE clearly visible'); drawnow;
[x,y] = ginput(2);
boundsB = round([min(x) max(x) min(y) max(y)]);
hold on;
plot(boundsB([1 2 2 1 1]),boundsB([3 3 4 4 3]),'r.-');
hold off;
pause(1);
close

%Save cropped image to new variable
B_zoom = imcrop(A, ...
    [boundsB(1) boundsB(3) (boundsB(2)-boundsB(1)) (boundsB(4)-boundsB(3))]);

%select a clear blue section
boundsB = [];
figure, imagesc(B_zoom);
axis image off;
title('Select a section representative of BLUE'); drawnow;
[x,y] = ginput(2);
boundsB = round([min(x) max(x) min(y) max(y)]);
hold on;
plot(boundsB([1 2 2 1 1]),boundsB([3 3 4 4 3]),'r.-');
hold off;
pause(1);
close

%Save cropped image to new variable
C{3} = imcrop(B_zoom, ...
    [boundsB(1) boundsB(3) (boundsB(2)-boundsB(1)) (boundsB(4)-boundsB(3))]);

%Calculate Ainv colour albedo matrix
cam_colors = []; proj_colors = []; projValue = 1;
for j = 1:3
  crop_C = C{j};
  crop_C = double(reshape(crop_C,[],3,1)');
  crop_P = zeros(size(crop_C));
  cam_colors = [cam_colors crop_C];
  if j == 1
     crop_P(1,:) = projValue;
  elseif j == 2
     crop_P(2,:) = projValue;
  else
     crop_P(3,:) = projValue;
  end
  proj_colors = [proj_colors crop_P];
```

```
end

Ainv = inv(cam_colors/proj_colors);

%Apply colour correction with this new Ainv value
%Allocate memory
COL = zeros(3,1);

%Apply colour correction to cropped image section only
for i = bounds(3):bounds(4)
    for j = bounds(1):bounds(2)
        COL(1) = I(i,j,1);
        COL(2) = I(i,j,2);
        COL(3) = I(i,j,3);

        rgb = Ainv*COL;

        for k = 1:3
            if rgb(k) <= 0
                rgb(k) = 0;
            elseif rgb(k) >= 1
                rgb(k) = 1;
            end
        end

        I(i,j,1) = rgb(1);
        I(i,j,2) = rgb(2);
        I(i,j,3) = rgb(3);
    end
end

%Save colour corrected image data
B = imcrop(I, ...
    [bounds(1) bounds(3) (bounds(2)-bounds(1)) (bounds(4)-bounds(3))]);

imshow(B); title('Colour corrected image')
fprintf('Imange Cleanup paused. Press enter to continue.\n');
pause;
%% Noise filtering
disp('+ Applying Non-linear means filter...')

%Crop out the bounded section for filtering and convert to 'double'
C = imcrop(I, ...
    [bounds(1) bounds(3) (bounds(2)-bounds(1)) (bounds(4)-bounds(3))]);

%Load path where noise filtering function is stored
addpath(['/Users/jacoverster/Documents/MATLAB/Add-Ons/'...
    'toolbox_nlmeans_version2']);

%Noise filtering function settings
Options.kernelratio = 4;
Options.windowratio = 4;
Options.verbose = true;
Options.filterstrength = 0.1;

%Filter operation
J = NLMF(C,Options); %NLMF function available from:
%https://www.mathworks.com/matlabcentral/fileexchange/...
%.../27395-fast-non-local-means-1d--2d-color-and-3d

%Write cropped and filtered data back into image
for y = bounds(1):bounds(2)
    for x = bounds(3):bounds(4)
        I(x,y,:) = J(x-bounds(3)+1,y-bounds(1)+1,:);
    end
end

%% Remove dark pixels
threshold = 0.03;
disp(['+ Removing dark pixels (with threshold = '...
    ,num2str(threshold),')...'])

%Allocate memory for variables
S = size(I);
DARK = zeros(S(1),S(2));
GRAY = rgb2gray(I);
GRAY_d = double(GRAY);
```

```
MAX = max(max(GRAY_d));
MIN = min(min(GRAY_d));

for j = bounds(1):bounds(2)
    for i = bounds(3):bounds(4)
        pn = (GRAY_d(i,j) - MIN)/(MAX - MIN);
        if pn < threshold
           DARK(i,j) = 1;
           I(i,j,:) = NaN;
       end
    end
end

%% Display results
disp('+ Displaying results...')
figure,
subplot(2,2,1),imshow(A); title('Original image')
subplot(2,2,2),imshow(B); title('Colour corrected image')
subplot(2,2,3),imshow(J); title('Non-linear means filtered image')
subplot(2,2,4),imshow(DARK(bounds(3):bounds(4),bounds(1):bounds(2)));...
    title('Dark pixels that were removed')

%% Save data
disp('+ Saving data...')
if ~exist('savefile','var')
    % second parameter does not exist, so default it to something
    savefile = 0;
end

if savefile == 1
    %save the image as a .png file
    imwrite(I, ['cleaned_up_image_(',imagename,').png']);
    %save the image, cropped image and bounds data in a .mat file
    Icrop = imcrop(I, ...
        [bounds(1) bounds(3) (bounds(2)-bounds(1)) (bounds(4)-bounds(3))]);
    save(['cleaned_up_image_(',imagename,').mat'], 'I', 'Icrop','bounds');
else
    %do nothing
end

disp('++ imageCleanup Done!')
```

```
function edgeMatching(imagename, orientation, savefile)
% This is the main edge matching algorithm
%
%Inputs: current_deB_I.mat (can be created manually from this script) and
%cleaned_up_image_(capture1.JPG).mat that is saved from imageCleanup.m
%
%Output: "matched_cam_edges" and "matched_proj_edges" are the final answers
%where each detected edge pixel in the camera image is mathced to a
%projected plane in the projector image. "matched_cam_edges" contains NaN
%for theses unmathced pixels.
%
%Created by: Jaco Verster (versterrie@gmail.com)

%Clear the workspace variables and close everything
%clc, clear , close all

%imagename = '2017_04_20Jaco2.JPG';

%Select calibration image based on portrait/landscape setup
if orientation == 'P'
    deBruijnImagename = 'debruijn_image_800x600x7.png'; %portrait
elseif orientation == 'L'
    deBruijnImagename = 'debruijn_image_600x800x7.png'; %landscape
else
    disp('error selecting calibration image')
end

disp('+++ edgeMatching initialized...')

%% Load images into memory, separate colour channels and run edge detection
disp('+ Detecting edges...')
%The part below can be commented out if current_deB_I.mat is up to date
```

```
%{
%the projected image
deBruijnImagename = 'debruijn_image_600x800x7.png';
Iproj = im2double(imread(deBruijnImagename));

%Seperate colour channels for projected image (only the first row is needed
%beacause all rows should be identical)
Rproj = Iproj(1,:,1);
Gproj = Iproj(1,:,2);
Bproj = Iproj(1,:,3);

%Size the image
S_proj = size(Iproj); %[row col]

%detect edges using detect_that_edge.m
edges_proj = edgeDetection(Iproj);

%Store deBruijn image information for fututure reference
save([deBruijnImagename,'.mat'],'edges_proj','Iproj','Rproj','Gproj','Bproj','S_proj')
%}

%load projector image data - NOTE: this must be commented out if the
%current_deB_I.mat file is out of date
load([deBruijnImagename,'.mat'])

%load the camera image
load(['cleaned_up_image_(',imagename,').mat']) %data from imageCleanup.m
Icam = Icrop;

%Seperate colour channels for camera image
Rcam = Icam(:,:,1);
Gcam = Icam(:,:,2);
Bcam = Icam(:,:,3);

%Size the image
S_cam = size(Icam); %[row col]

%detect edges using detect_that_edge.m
edges_cam = edgeDetection(Icam);

%% Get edge gradient values for the camera image and count the nr of edges
%compute 1-D gradient for each colour band
grad_cam_R = gradient(Rcam);
grad_cam_G = gradient(Gcam);
grad_cam_B = gradient(Bcam);

%Scan through the camera image & create a edge location matrix
for i = 1:S_cam(1)
    counter = 0;
    for j = 2:S_cam(2)-1
        %compile the edge matrix with associated gradient values
        if edges_cam(i,j) == 1
            counter = counter + 1;
            edge_grad_cam_R(i,counter) = max(grad_cam_R(i,j-1:j+1));
            edge_grad_cam_G(i,counter) = max(grad_cam_G(i,j-1:j+1));
            edge_grad_cam_B(i,counter) = max(grad_cam_B(i,j-1:j+1));
            %max function above is used to select the highest of 3 values
            %around the detected edge - this is to account for a small
            %offset in the R, G and B peak locations that is due
            %to the fact that the R, G, B detectors are not in exactly the
            %same location on the camera

            %create a edge location matrix for the camera
            loc_cam_edges(i,counter) = j;
            nr_of_cam_edges(i) = counter;
        end
    end
end

%% Determine the location of the detected edges inside the deBruijn pattern
disp('+ Determining edge location inside deBruin pattern...')
%Determine the average distance between edges for loc_cam_edges in the
%edge_matching script

%Convert all zero values in loc_cam_edges to NaN for ease of computation
loc_cam_edges(loc_cam_edges==0) = NaN;
```

```
%Calculate edge distance
a = diff(loc_cam_edges,1,2); %a(row,1) => loc_cam_edges(row,1->2);

%calculate the mean edge distance
b = permute(a,[2 1]);
mean_edge_dist = tsnanmean(b)';

%max number of detected camera edges
max_edges = max(nr_of_cam_edges);

%Manipulate loc_cam_edges to place undetected edges in the correct location
%and to remove false positives
for row = 1404:S_cam(1)
    skipped_cols = find(a(row,:) > 1.5*mean_edge_dist(row));
    if isempty(skipped_cols) == 0
        count = 1;
        for s = 1:length(skipped_cols)
            working_col = skipped_cols(s)+count;
            move = loc_cam_edges(row,working_col:max_edges-1);
            loc_cam_edges(row,working_col) = nan;
            loc_cam_edges(row,working_col+1:max_edges) = move;
            count = count+1;
        end
    end
end

%Convert all zero values in loc_cam_edges to NaN again
loc_cam_edges(loc_cam_edges==0) = NaN;

%re-calculate edge distances, mean and min edge distance per row
a2 = diff(loc_cam_edges,1,2);
b2 = permute(a2,[2 1]);
mean_edge_dist2 = tsnanmean(b2)';
min_edge_dist = min(b2)';

%Pattern used in the deBruijn image generator: deBruijnImage.m
pattern = [1,2,5,4,3,5,4,3,4,3,0,1,2,5,3,5,4,3,4,1,4,3,4,3,0,2,5,4,3,0,...
    3,4,3,5,4,1,2,5,3,4,5,4,3,5,3,5,4,3,5,3,0,1,2,1,4,1,2,5,3,0,3,4,3,0,...
    1,0,1,2,0,1,4,3,4,5,3,5,4,3,0,2,0,1,2,0,3,5,4,3,0,3,0,1,0,1,0,2,5,3,...
    4,1,4,3,5,4,5,3,4,1,2,0,3,4,1,0,1,4,3,5,3,5,3,4,1,0,2,0,1,0,2,0,2,5,...
    3,5,2,0,1,0,3,4,5,4,1,4,1,0,1,0,3,5,2,5,3,0,3,5,4,1,4,5,2,5,2,5,2,1,...
    2,0,1,4,5,4,5,3,5,3,4,5,3,5,2,5,2,1,4,1,2,0,2,0,3,4,5,2,1,0,1,4,5,3,...
    0,1,4,5,2,1,2,0,3,0,3,5,3,5,3,5,2,1,0,2,0,3,5,3,0,2,0,2,1,4,5,2,1,4,...
    1,4,5,3,0,3,0,3,0];

%Find the first set of 5 consecutive detected edges so that we can
%determine the x-loc in the deBruijn pattern.

conseqEdge = zeros(S_cam(1),7); %allocate memory for variable
%%%%%%%%%%%%%%%%%%conseqEdge = zeros(1,7); %do the first row only

for row = 1:S_cam(1)
    col = 1;
    while col <= (size(a2,2)-4) % changed from (S_cam(2)-4)
        dif = diff(a2(row,col:(col+4)));
        val = a2(row,col);
        val2 = isnan(dif);
        val3 = abs(max(dif)-min(dif));
        if val <= mean_edge_dist2(row)+4 &&...%check upper limit
                val >= mean_edge_dist2(row)-4 &&... %check lower limit
                loc_cam_edges(row,col) > mean_edge_dist2(row)/1.8 &&...
                any(val2) == 0 &&...
                val3 <= min_edge_dist(row);
                %check that first edge is far enough from the edge of Icam
                %and that there are no NaN values (val2)
                %and that the edges are not too far apart (val3)
            %Save the consecutive edge data as follows:
            %column 1 = edge number
            conseqEdge(row,1) = col;
            %column 2 = median value of edge distance (in pixels)
            conseqEdge(row,2) = mean_edge_dist2(row);
            %column 3:7 = the edge pixel numbers in camera_crop image...
            %(remember to add the "bounds" values! ==> [col + bounds(1)-1])
            conseqEdge(row,3:7) = loc_cam_edges(row,col:(col+4));
            col = S_cam(2);
        end
        col = col+1;
```

```
    end
end

%Extract the RGB values from the camera image on either side of each edge
%detected above
RGB = cell(S_cam(1),6);
%%%%%%%%%%%%%%%%%%%RGB = cell(1,6);
counter = 0;
for row = 1:S_cam(1)
    if conseqEdge(row,1) ~= 0
        RGB{row,1}(1:3) = Icam(row,round(conseqEdge(row,3)-...
            conseqEdge(row,2)/2),:);
        RGB{row,2}(1:3) = Icam(row,round(conseqEdge(row,4)-...
            conseqEdge(row,2)/2),:);
        RGB{row,3}(1:3) = Icam(row,round(conseqEdge(row,5)-...
            conseqEdge(row,2)/2),:);
        RGB{row,4}(1:3) = Icam(row,round(conseqEdge(row,6)-...
            conseqEdge(row,2)/2),:);
        RGB{row,5}(1:3) = Icam(row,round(conseqEdge(row,7)-...
            conseqEdge(row,2)/2),:);
        RGB{row,6}(1:3) = Icam(row,round(conseqEdge(row,7)+...
            conseqEdge(row,2)/2),:);
    else
        counter = counter+1;
    end
end

%Identify correct colour and name colour 0 to 5 according to the following
%convention - this is the sam convention used for the deBruijn pattern
colM =  [1 1 0; %yellow = 3
        1 0 1;  %magenta = 4
        0 1 1;  %cyan = 5
        1 0 0;  %red = 2
        0 1 0;  %green = 1
        0 0 1]; %blue = 0

RGB_pattern = nan(S_cam(1),6); %allocate memory
%%%%%%%%%%%%%%%%%%RGB_pattern = nan(1,6);
col_threshold = 0.3;  %%%THIS THRESHOLD VALUE IS VERY IMPORTANT
for row = 1:S_cam(1)
    for i = 1:6
        %Extract colour information from cell and apply thresholding
        A = RGB{row,i};
        col_max = max(A); %records the maximum value across colour channels
        if isempty(A) == 0
            for j = 1:3
                if A(j) > col_threshold*col_max
                    RGB{row,i}(j) = 1;
                else
                    RGB{row,i}(j) = 0;
                end
            end
            %Match colour and rename
            if RGB{row,i} == colM(1,:)
                RGB_pattern(row,i) = 3;
            elseif RGB{row,i} == colM(2,:)
                RGB_pattern(row,i) = 4;
            elseif RGB{row,i} == colM(3,:)
                RGB_pattern(row,i) = 5;
            elseif RGB{row,i} == colM(4,:)
                RGB_pattern(row,i) = 2;
            elseif RGB{row,i} == colM(5,:)
                RGB_pattern(row,i) = 1;
            elseif RGB{row,i} == colM(6,:)
                RGB_pattern(row,i) = 0;
            end
        end
    end
end

%Search for the above patterns in the deBruijn image
loc_in_deBruijn = nan(S_cam(1),1); %allocate memory
%%%%%%%%%%%%%%%%%%loc_in_deBruijn = [];
nA = numel(pattern); %calculate deBruijn pattern length
np = 6; %search pattern length
%Search the big pattern for the small pattern match
for row = 1:S_cam(1)
```

```
    p = RGB_pattern(row,:);
    buffer = ~any(spdiags(repmat(pattern(:), 1, np), 0:np-1, nA, nA) - ...
                spdiags(repmat(p, nA, 1), 0:np-1, nA, nA), 2);
    loc = find(full(buffer(1:nA-np+1)));
    if loc > 0
        loc_in_deBruijn(row) = loc(1); %This records the first stripe nr
    end
end

%% Get edge gradient values for the projector image and count nr of edges

%compute 1-D gradient for each colour band
%'/0.5' is used to get values of -1 and 1 for projected image
grad_proj_R = gradient(Rproj)/0.5;
grad_proj_G = gradient(Gproj)/0.5;
grad_proj_B = gradient(Bproj)/0.5;

%mark the number of edges to match for the projected image
nr_of_proj_edges = max_edges;

%create an edge location array for the projector
loc_proj_edges = find(edges_proj(1,:) == 1);

%allocate memory for matched projector edges
matched_proj_edges = NaN([S_cam(1) nr_of_proj_edges],'double');
edge_grad_proj_R = NaN([S_cam(1) nr_of_proj_edges],'double');
edge_grad_proj_G = NaN([S_cam(1) nr_of_proj_edges],'double');
edge_grad_proj_B = NaN([S_cam(1) nr_of_proj_edges],'double');

%Initialize  unmatched_edges variable
unmatched_edges = 0;
skipped_rows = [];

%compile the edge matrix with associated gradient values
for i = 1:S_cam(1)
    %first check if loc_in_deBruijn exists, if not skip row and make note
    if isnan(loc_in_deBruijn(i)) == 0
        %calculate the first term according to the mathced deB pattern
        matched_loc_in_deB = loc_in_deBruijn(i)+1-conseqEdge(i,1);

        %store matched projector edges starting from calculated match up to
        %max_edges
        from = matched_loc_in_deB;
        to = min(length(loc_proj_edges),...
            matched_loc_in_deB + nr_of_proj_edges - 1);
        nr_of_matched_edges(i) = to - from + 1;

        matched_proj_edges(i,1:nr_of_matched_edges(i)) = loc_proj_edges(from:to);

        %compile the edge matrix
        edge_grad_proj_R(i,1:nr_of_matched_edges(i)) = ...
            grad_proj_R(matched_proj_edges(i,1:nr_of_matched_edges(i)));
        edge_grad_proj_G(i,1:nr_of_matched_edges(i)) = ...
            grad_proj_G(matched_proj_edges(i,1:nr_of_matched_edges(i)));
        edge_grad_proj_B(i,1:nr_of_matched_edges(i)) = ...
            grad_proj_B(matched_proj_edges(i,1:nr_of_matched_edges(i)));
    else
        unmatched_edges = unmatched_edges + nr_of_cam_edges(i);
        skipped_rows = [skipped_rows; i];
    end
end
disp(['NOTE: ',num2str(length(skipped_rows)),...
    ' rows skipped because no loc_in_deBruijn found, see "skipped_rows"'])

%% Match projector and camera edges
disp('+ Matching projector and camera edges...')
%allocate memory for variables
matched_cam_edges = NaN([S_cam(1) nr_of_proj_edges],'double');

%scan through all the detected camera edges and compute the matching score
%remove skipped rows from scan
A = 1:S_cam(1); %all rows
A(skipped_rows) = []; %remove skipped rows

%This is where the edge matching process happens
for row = A
    %allocate memory for score
```

```
score = zeros(max(nr_of_cam_edges(row))+3,nr_of_proj_edges+3);
%Construct the score matrix for each row
for ec = 1:nr_of_cam_edges(row) %length(cam_edge_grad_R(2,:))
    for ep = 1:nr_of_matched_edges(row)
        %Compute match score matrix for each row
        score(ec,ep) = min([...
    consistency(edge_grad_proj_R(row,ep),edge_grad_cam_R(row,ec)),...
    consistency(edge_grad_proj_G(row,ep),edge_grad_cam_G(row,ec)),...
    consistency(edge_grad_proj_B(row,ep),edge_grad_cam_B(row,ec))]);
    end
end

%{
%determine the starting location for the optimal path
%NOTE: this functionality is no longer needed due to the way
%loc_cam_edges is constructed - start should always be [1 1].
[row_1_max,c] = max(score(1,1:4));
[col_1_max,r] = max(score(1:3,1));
if row_1_max == col_1_max
    start = [1,1];
    edges_matched = 1;
elseif row_1_max > col_1_max %&& score(2,c+1) > score(r+1,2)
    start = [1,c];
    edges_matched = 2;
elseif row_1_max < col_1_max %&& score(2,c+1) < score(r+1,2)
    start = [r,1];
    edges_matched = 1;
else
disp('ERROR!')
end
%}

%Now calculate optimal path with a set threshold
threshold = 0.001;

%This is the starting position
start = [1,1];
i = start(1);
j = start(2);

%Initialize edge matching and match first edge
edges_matched = 1;
matched_cam_edges(row,edges_matched) = loc_cam_edges(row,i);

%Scan through the score matrix and do edge matching
while i < nr_of_cam_edges(row) && j < nr_of_proj_edges
    if score(i+1,j+1) > threshold
        score(i+1,j+2)=0; score(i+1,j+3)=0;
        score(i+2,j+1)=0; score(i+3,j+1)=0;
        i=i+1;
        j=j+1;
        edges_matched = edges_matched + 1;
        matched_cam_edges(row,edges_matched) = ...
            loc_cam_edges(row,edges_matched);
    elseif score(i+1,j+2) > threshold
        score(i+1,j+3)=0; score(i+2,j+1)=0; score(i+3,j+1)=0;
        i=i+1;
        j=j+2;
        edges_matched = edges_matched + 2;
        unmatched_edges = unmatched_edges+1;
        matched_cam_edges(row,edges_matched) = ...
            loc_cam_edges(row,edges_matched);
    elseif score(i+2,j+1) > threshold
        score(i+1,j+3)=0; score(i+3,j+1)=0;
        i=i+2;
        j=j+1;
        edges_matched = edges_matched + 1;
        matched_cam_edges(row,edges_matched) = ...
            loc_cam_edges(row,edges_matched);
    elseif score(i+1,j+3) > threshold && score(i+2,j+4) > threshold
        score(i+3,j+1)=0;
        i=i+1;
        j=j+3;
        edges_matched = edges_matched + 3;
        unmatched_edges = unmatched_edges+2;
        matched_cam_edges(row,edges_matched) = ...
            loc_cam_edges(row,edges_matched);
```

```
        elseif score(i+3,j+1) > threshold && score(i+4,j+2) > threshold
            i=i+3;
            j=j+1;
            edges_matched = edges_matched + 1;
            matched_cam_edges(row,edges_matched) = ...
                loc_cam_edges(row,edges_matched);
        else
            i=i+1;
            j=j+1;
            edges_matched = edges_matched + 1;
            matched_cam_edges(row,edges_matched) = NaN;
            unmatched_edges = unmatched_edges+1;
            %Print out unmatched edge locations
            display(num2str([row j]),'Unmatched edge at [row,col]')
        end
    end
    %display(['row ',num2str(row),' done']) %To track progress
end

%% Verify matching
%
disp('+ Plotting edge matching for final visual verification...')
%display the percentage of edge pixels in the camera that are unmatched
Percentage_unmatched = 100*unmatched_edges/numel(matched_cam_edges)

%Plot a few matched locations on both proj and cam images to visually
%verify good matching
cam_win = 25;
proj_win = 10;
marker1 = 0.25;
marker2 = 0.75;

plotRow_1 = round(min(S_cam(1),S_proj(1))*marker1);
plotRow_2 = round(min(S_cam(1),S_proj(1))*marker2);

plotColCam_1 = matched_cam_edges(plotRow_1,round(max_edges*marker1));
plotColCam_2 = matched_cam_edges(plotRow_1,round(max_edges*marker2));
plotColCam_3 = matched_cam_edges(plotRow_2,round(max_edges*marker1));
plotColCam_4 = matched_cam_edges(plotRow_2,round(max_edges*marker2));
plotColProj_1 = matched_proj_edges(plotRow_1,round(max_edges*marker1));
plotColProj_2 = matched_proj_edges(plotRow_1,round(max_edges*marker2));
plotColProj_3 = matched_proj_edges(plotRow_2,round(max_edges*marker1));
plotColProj_4 = matched_proj_edges(plotRow_2,round(max_edges*marker2));

Cam_1 = Icam((plotRow_1-cam_win):(plotRow_1+cam_win),...
    (plotColCam_1-cam_win):(plotColCam_1+cam_win),:);
Cam_2 = Icam((plotRow_1-cam_win):(plotRow_1+cam_win),...
    (plotColCam_2-cam_win):(plotColCam_2+cam_win),:);
Cam_3 = Icam((plotRow_2-cam_win):(plotRow_2+cam_win),...
    (plotColCam_3-cam_win):(plotColCam_3+cam_win),:);
Cam_4 = Icam((plotRow_2-cam_win):(plotRow_2+cam_win),...
    (plotColCam_4-cam_win):(plotColCam_4+cam_win),:);

Proj_1 = Iproj((plotRow_1-proj_win):(plotRow_1+proj_win),...
    (plotColProj_1-proj_win):(plotColProj_1+proj_win),:);
Proj_2 = Iproj((plotRow_1-proj_win):(plotRow_1+proj_win),...
    (plotColProj_2-proj_win):(plotColProj_2+proj_win),:);
Proj_3 = Iproj((plotRow_2-proj_win):(plotRow_2+proj_win),...
    (plotColProj_3-proj_win):(plotColProj_3+proj_win),:);
Proj_4 = Iproj((plotRow_2-proj_win):(plotRow_2+proj_win),...
    (plotColProj_4-proj_win):(plotColProj_4+proj_win),:);

%Plot matched image sections
figure('Name','Matched camera and projector edges compared visually')
subplot(2,4,1),imshow(Cam_1); title('Cam_1')
subplot(2,4,2),imshow(Cam_2); title('Cam_2')
subplot(2,4,3),imshow(Cam_3); title('Cam_3')
subplot(2,4,4),imshow(Cam_4); title('Cam_4')
subplot(2,4,5),imshow(Proj_1); title('Proj_1')
subplot(2,4,6),imshow(Proj_2); title('Proj_2')
subplot(2,4,7),imshow(Proj_3); title('Proj_3')
subplot(2,4,8),imshow(Proj_4); title('Proj_4')
%}
%% Show matched edges
Iedges = Icam;
for i = 1:S_cam(1)
    for j = 1:max_edges
```

```
        col_val = matched_cam_edges(i,j);
        if isnan(col_val) == 0
            Iedges(i,col_val,1)=255;
            Iedges(i,col_val,2)=0;
            Iedges(i,col_val,3)=0;
        end
    end
end
figure('Name','Matched edges shown on image'), imshow(Iedges)

%% Save
disp('+ Saving data...')
%The relevant data is stored in matched_edges.mat
%savefile = input...
    %('Do you want to save the matched edges? [1 = yes, other = no]: ','s');
if savefile == 1
    save(['matched_edges_(',imagename,').mat'],'matched_cam_edges',...
        'matched_proj_edges','Rcam','Gcam','Bcam')
    display('Results SAVED.')
else
    display('Results were NOT saved.')
end

disp('++ edgeMatching done!')
```

```
function noiseless_EDGE = edgeDetection(I)
%This is the main edge detection algorithm used
%Input: Image (.JPG)
%Output: EDGE2 variable with detected edges and 2 images plotted to show
%the results visually
%
%Created by: Jaco Verster (versterrie@gmail.com)

%Manual override
%imagename = 'canoncrop.jpg';
%I = im2double(imread(imagename));
%figure, imshow(I)

R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);

%create empty variables
S = size(I); %[row col]
EDGE = zeros(S(1),S(2));
GRADmx = [];

%Scan through each row and find local maxima using findpeaks.m
for i = 1:S(1)
    %compute the 1D gradient for each color channel
    grad_R = gradient(R(i,:));
    grad_G = gradient(G(i,:));
    grad_B = gradient(B(i,:));

    %compute the sum of the squares of all thre channels
    GRAD = grad_R.^2 + grad_G.^2 + grad_B.^2;
    GRADmx(i,:) = GRAD; %store the gradient calculation

    %detecting local maximas with PEAK = peakdet(GRAD,delta); was updated
    %to Matlab's built in findpeaks function on 02/02/2017 to minimize
    %external functions - function is slightly faster than findpeaks for
    %low values of delta

    %Peak detection is done here
    %[pks, locs] = findpeaks(GRAD,'MinPeakDistance',stripewidth-5,...
    %    'MinPeakProminence',0.001);
    [pks, locs] = findpeaks(GRAD,'MinPeakProminence',0.001,...
        'MinPeakDistance',10);

    PEAK = [locs(:), pks(:)];

    if prod(size(PEAK)) ~= 0
        %mark how many peaks were detected
        PEAK_l = length(PEAK(:,1));
```

```
        %make note of the minimum peak detected
        PEAK_min(i) = min(PEAK(:,2));
        %calculate PEAK gradient to view edge distance in pixels
        dPEAK = diff(PEAK);

        %write all peak values to EDGE
        for k = 1:PEAK_l
            EDGE(i,PEAK(k,1)) = 1;
        end
        %{
        %compare two local maximas that are too close and erase the smaller
        for k = 1:PEAK_l-1
            if dPEAK(k) < stripewidth-1
               [loc_max,index] = max(PEAK(k:k+1,2));
               EDGE(i,PEAK(k,1)) = 0;
               EDGE(i,PEAK(k+1,1)) = 0;
               EDGE(i,PEAK(k-1+index,1)) = 1;
               PEAK(k+2-index,2) = 0;
            end
        end

        %Mark rows for which all the edges are not detected and show how many
        %edges were detected
        if PEAK_l < expected_edges && PEAK_l > 0
            rows_with_less_edges = rows_with_less_edges + 1;
            WARNINGS(i) = PEAK_l;
        elseif PEAK_l > expected_edges
            rows_with_more_edges = rows_with_more_edges + 1;
            WARNINGS(i) = PEAK_l;
        end
        %}
    end

    %clear key variables for next iteration
    clear PEAK, clear dPEAK, clear GRAD
    clear grad_R, clear grad_G, clear grad_B;
end

%Clean up the edge data by removing isolated pixels from EDGE data: only
%allow pixels that are connected to 10 or more other pixels
noiseless_EDGE = bwareaopen(EDGE,10);
%{
%CC = bwconncomp(EDGE2); %count number of connected objects

%display edge detection information
%rows_with_more_edges
%rows_with_less_edges
%display(imagename)
%number_of_detected_objects = CC.NumObjects

%Plot local maximas in the middel of the image to confirm good PEAK
%detection
%figure, plot(GRADmx(round(S(1)/2),:),'k')
%}

%before isolated pixel removal
%figure, imshow(EDGE)

%after isolated pixel removal
%figure('Name','Detected edges'), imshow(noiseless_EDGE)

%Show detected edges
Iedges = I;
for i=1:S(1)
    for j=1:S(2)
        if noiseless_EDGE(i,j) == 1
            Iedges(i,j,1)=255; Iedges(i,j,2)=0; Iedges(i,j,3)=0;
        end
    end
end

figure('Name','Detected edges shown on image'), imshow(Iedges)
```

```
function edgeTriangulation(imagename,savefile)
```

```
% This is the main triangulation algorithm for creating a pointcloid
%
%Created by: Jaco Verster (versterrie@gmail.com)

%Clear the workspace variables and close everything
%clc, clear , close all

%imagename = '170411deB-P2.JPG';

disp('+++ edgeTriangulation initialized...')

%% Load all the required  toolbox paths and data
disp('+ Loading data and toolboxes...')
%Paths for external toolboxes used:
%[1] Jean-Yves Bouguet, "Camera Calibration Toolbox for Matlab",
    %http://www.vision.caltech.edu/bouguetj/calib_doc/, and
%[2] G. Falcao, et al., "Projector-Camera Calibration Toolbox",
    %http://code.google.com/p/procamcalib, 2009. In one folder.
addpath('/Users/jacoverster/Documents/MATLAB/Cam_Proj_Calib')

%[3] Douglas Lanman, "Structured Light for 3D Scanning",
    %http://mesh.brown.edu/byo3d/index.html
addpath('/Users/jacoverster/Documents/MATLAB/mlStructuredLight')

%Load camera-projector calibration values
load('Cam_Proj_calib.mat')

%Load edge data from edgeMatching.m
load(['matched_edges_(',imagename,').mat'])
%Load crop-boundary values for cleaned up image
load(['cleaned_up_image_(',imagename,').mat'])
%Note: 'bounds' variable format => [x1 x2 y1 y2]

%% Determine mapping from projector pixels to optical rays
if exist('Op') == 0 %Check if Op has be calculated for the camera before.
    disp('+ Computing plane equations for projected edges...')
    c = 1:nx_proj;
    r = 1:ny_proj;
    [C,R] = meshgrid(c,r);

    np  = pixel2ray([C(:) R(:)]',fc_proj,cc_proj,kc_proj,alpha_c_proj);

    %%%translate to camera space unsig XXc = Rc_ext * XX + Tc_ext from
    %http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.htmll
    np = R_proj'*(np - T_proj*ones(1,size(np,2)));

    %Reshape data to a matrix format that matches the Projector dimensions
    Np = zeros([ny_proj nx_proj 3]);
    Np(:,:,1) = reshape(np(1,:),ny_proj,nx_proj);
    Np(:,:,2) = reshape(np(2,:),ny_proj,nx_proj);
    Np(:,:,3) = reshape(np(3,:),ny_proj,nx_proj);

    %Calculate the center of the projector in camera coordinates
    Op = -R_proj'*T_proj;
    save(['Cam_Proj_calib.mat'],'Np','Op','-append')

elseif exist('Np') == 1
    disp('+ Plane equations for projected edges computed previously...')
end
%% Estimate plane equations describing every projector column
% Notes:
% - Resulting coefficient vector is in camera coordinates.
% - fitPlane.m tested and confirmed using an online calculator:
%   http://www.ambrsoft.com/TrigoCalc/Plan3D/Plane3D_.htm

%Estimate the plane equations describing every projector column
wPlaneCol = zeros(nx_proj,4);
for col = 1:nx_proj
   wPlaneCol(col,:) = fitPlane(...
       [Op(1); Np(:,col,1)],[Op(2); Np(:,col,2)],[Op(3); Np(:,col,3)]);
   %figure(4); hold on;
   %plot3(Np(:,col,1),Np(:,col,3),-Np(:,col,2),'r-');
   %drawnow;
end

%Estimate the plane equations describing every projector row
wPlaneRow = zeros(ny_proj,4);
```

```
for i = 1:ny_proj
   wPlaneRow(i,:) = ...
      fitPlane([Op(1) Np(i,:,1)],[Op(2) Np(i,:,2)],[Op(3) Np(i,:,3)]);
   %figure(4); hold on;
   %plot3(Np(i,:,1),Np(i,:,3),-Np(i,:,2),'g-');
   %drawnow;
end

%% Compute optical rays for each camera pixel
if exist('Nc') == 0 %Check if Nc has be calculated for the camera before.
    disp('+ Computing ray-equations for camera pixels...')
    c = 1:nx_cam;
    r = 1:ny_cam;
    [C,R] = meshgrid(c,r);

    nc = pixel2ray([C(:) R(:)]'-1,fc_cam,cc_cam,kc_cam,alpha_c_cam);
    %%%not sure what this '-1' is for - Needs investigating

    %Re-shape Nc
    Nc = zeros([ny_cam nx_cam 3]);
    Nc(:,:,1) = reshape(nc(1,:),ny_cam,nx_cam);
    Nc(:,:,2) = reshape(nc(2,:),ny_cam,nx_cam);
    Nc(:,:,3) = reshape(nc(3,:),ny_cam,nx_cam);

    %Camera center
    Oc = [0; 0; 0];
    save(['Cam_Proj_calib.mat'],'Nc','Oc','-append')

elseif exist('Nc') == 1
    disp('+ Ray-equations for camera pixels computed previously...')
end

%% Reconstruct 3D points using intersection with illumination plane(s)
disp('+ Computing ray-plane intersections reconstructing 3D points...')
%%%%%%%%%%
%intersectLineWithPlane.m tested and confirmed using an online caluclator:
%http://www.ambrsoft.com/TrigoCalc/Plan3D/PlaneLineIntersection_.htm
%%%%%%%%%%
%Check if "I" (camera image) is landscape or portrait and select
%oppropriate reconstruction method
S_cam = size(I);

if S_cam(1) < S_cam(2) %Landscape method
    %Allocate memory
    vertices = zeros([size(matched_cam_edges),3]);
    colours   = vertices;

    for y = 1:length(matched_cam_edges(:,1))
        for x = 1:length(matched_cam_edges(1,:))
            if ~isnan(matched_cam_edges(y,x))
                cam_x = bounds(1) + matched_cam_edges(y,x);
                cam_y = y + bounds(3) - 1;
                proj_x = matched_proj_edges(y,x);

                CamPixel = Nc(cam_y,cam_x,:);
                ProjPlane = wPlaneCol(proj_x,:)';

                vertices(y,x,:) = intersectLineWithPlane(...
                    Oc,CamPixel(:),ProjPlane)';
                colours(y,x,1) = Rcam(y,matched_cam_edges(y,x));
                colours(y,x,2) = Gcam(y,matched_cam_edges(y,x));
                colours(y,x,3) = Bcam(y,matched_cam_edges(y,x));
            else
                colours(y,x,1) = NaN;
                colours(y,x,2) = NaN;
                colours(y,x,3) = NaN;
            end
        end
    end

elseif S_cam(1) > S_cam(2) %Portrait method
    %Swop row and col for around when going from matched_cam_edges to Nc
    %because calibration was done in landscape while edge detection was
    %done in portrait mode
    %Explanation: [e_row, e_col] =/= [2304-e_col, e_row],
    %[1346:2484, 603:1732] =/= [2304-603:1732, 1346:2484]
```

```
    %Allocate memory
    vertices = zeros([size(matched_cam_edges),3]);
    colours   = vertices;

    for y = 1:length(matched_cam_edges(:,1)) %bounds(3)&(4) in play
        for x = length(matched_cam_edges(1,:)):-1:1 %bounds(1)&(2) in play
            if ~isnan(matched_cam_edges(y,x)) && ...
                    ~isnan(matched_proj_edges(y,x))
                cam_row = ny_cam - (matched_cam_edges(y,x) + bounds(1));
                cam_col = y + bounds(3) - 1;
                proj_row = ny_proj - matched_proj_edges(y,x);
                CamPixel = Nc(cam_row,cam_col,:);
                ProjPlane = wPlaneRow(proj_row,:)';
                vertices(y,x,:) = intersectLineWithPlane(...
                    Oc,CamPixel(:),ProjPlane)';
                colours(y,x,1) = Rcam(y,matched_cam_edges(y,x));
                colours(y,x,2) = Gcam(y,matched_cam_edges(y,x));
                colours(y,x,3) = Bcam(y,matched_cam_edges(y,x));
            else
                colours(y,x,1) = NaN;
                colours(y,x,2) = NaN;
                colours(y,x,3) = NaN;
            end
        end
    end
end

%Make all zeo values = NaN
vertices(vertices==0) = NaN;

%% Display the point cloud
disp('+ Displaying point cloud...')

%Create a pointCloud class from points and color information
ptCloud = pointCloud(vertices,'Color',colours);
ptCloudValid = removeInvalidPoints(ptCloud); %remove invalid (NaN) points
ptCloudDenoise = pcdenoise(ptCloudValid,'NumNeighbors',25);

% Display the point cloud with per-vertex color.
%player = pcplayer(ptCloud.XLimits,ptCloud.YLimits,[480,560]);
%view(player,ptCloudDenoise)
figure, pcshow(ptCloudValid)
xlabel('x'), ylabel('y'), zlabel('z')

%% Save the pointCloud to a file
disp('+ Saving data...')
if savefile == 1;
    pcwrite(ptCloudValid,['pointCloud_',imagename])
    save(['matched_edges_(',imagename,').mat'],'matched_cam_edges',...
        'matched_proj_edges','Rcam','Gcam','Bcam','vertices')
    display('Results SAVED.')
else
    display('Results were NOT saved.')
end

disp('++ edgeTriangulation done!')

%more info about using pointclouds can be found at:
%http://www.geo.tuwien.ac.at/downloads/pg/pctools/publish/...
%.../pointCloudIntro/html/pointCloudIntro.html
```

```
function [gx,gy,gz,H,Pmax] = surfaceCurvature(imagename,pointcloud,plot)
% This function calculates the surface curvature from a set of points
%Inputs:
%   - imagename
%   - a pointcloud
%   - plot = 1/other - only produces plots if value is exactly 1
%
%Created by: Jaco Verster (versterrie@gmail.com)

%Extract vertices data and convert to vector
x = double(pointcloud.Location(:,1)); %double included to keep gridfit happy
y = double(pointcloud.Location(:,2));
z = double(pointcloud.Location(:,3));
```

```
%Create a grid to match the surface using gridfit
gx = double(pointcloud.XLimits(1)):4:double(pointcloud.XLimits(2));
gy = double(pointcloud.YLimits(1)):4:double(pointcloud.YLimits(2));
gz = gridfit(x,y,z,gx,gy); %,'regularizer','springs','smoothness',[0.1 1]);

%after experimenting with the above variation I could not really beat the
%"out of the box performance of gridfit - decided to keep settings 'default'

%Calculate surface curvature of the fitted grid
[X,Y] = meshgrid(gx,gy);
[K,H,Pmax,Pmin] = surfature(X,Y,gz); %surfature function available from:
%https://www.mathworks.com/matlabcentral/fileexchange/11168-surface-curvature

if plot == 1
    %Plot the surface and pointcloud to verify if a good fit was found
    figure
    colormap(hot(256));
    surf(-gx,gy,-gz); %mirror x and z coordinates to align
    camlight right;
    lighting phong;
    hold on
    scatter3(-x,y,-z,'.');
    title 'Gridfit surface and pointcloud data'
    xlabel('x'), ylabel('y'), zlabel('z')

    %plot the surface curvature
    figure
    surf(-X,Y,-gz,H,'facecolor','interp'); %mirror x and z coordinates to align
    title 'Surface curvature'
    xlabel('x'), ylabel('y'), zlabel('z')

    %calculate and plot surface normals
    figure
    surfnorm(X,Y,gz)
    title 'Surface normals'
    xlabel('x'), ylabel('y'), zlabel('z')
end

%write the gridfit data to an .stl mesh file
stlwrite(['pointSurface_',imagename,'.stl'],gx,gy,gz);
```

```
function createFigure(xdata1, ydata1, zdata1, cdata1, X3, Y3, Z3)
%CREATEFIGURE(XDATA1, YDATA1, ZDATA1, CDATA1, X3, Y3, Z3, C3)
%  XDATA1:  surface xdata
%  YDATA1:  surface ydata
%  ZDATA1:  surface zdata
%  CDATA1:  surface cdata
%  X3:  scatter3 x
%  Y3:  scatter3 y
%  Z3:  scatter3 z
%  C3:  scatter3 c

%  Auto-generated by MATLAB on 09-Nov-2017 11:12:44

% Create figure
figure1 = figure('InvertHardcopy','off','Color',[1 1 1]);
colormap('jet');

% Create axes
axes1 = axes;
hold(axes1,'on');

% Create surf
surf(xdata1,ydata1,zdata1,cdata1,'FaceColor','interp');

% Create scatter3
%scatter3(X1,Y1,Z1,'*','k');

% Create scatter3
%scatter3(X2,Y2,Z2,'o','r');

% Create scatter3
scatter3(X3,Y3,Z3,'*','k');
```

```
% Create xlabel
xlabel({'x [mm]'},'FontWeight','bold','FontSize',16,'FontName','calibri');

% Create zlabel
zlabel({''},'FontWeight','bold','FontSize',16,'FontName','calibri');

% Create ylabel
ylabel({'y [mm]'},'FontWeight','bold','FontSize',16,'FontName','calibri');

%indicate figure size
set(gcf,'units','pixels','position',[270,1065,723,371])

view(axes1,[-180 90]);
grid(axes1,'on');
% Set the remaining axes properties
set(axes1,'CLim',[-0.03 0.03],'DataAspectRatio',[1 1 1],'FontName',...
    'calibri','FontSize',16,'OuterPosition',[0 0 0.626964878739968 1],...
    'PlotBoxAspectRatio',[2.16736975575005 1.64017170705409 1]);
% Create colorbar
colorbar('peer',axes1,'Position',...
    [0.58429690662011 0.149922720247295 0.0172655933798905 0.731066460587326],...
    'FontSize',16);

% Create textbox
annotation(figure1,'textbox',...
    [0.570109952628629 0.0656250000000001 0.0603587973713713 0.0663174056527917],...
    'String',{'Mean curv.','[1/mm]'},...
    'Margin',1,...
    'HorizontalAlignment','center',...
    'FontWeight','bold',...
    'FontSize',16,...
    'FontName','Calibri',...
    'FitBoxToText','off');
```

7.2 Appendix II: Informed Consent Form and Research Protocol

PARTICIPANT INFORMATION SHEET AND INFORMED CONSENT FORM

FACULTY OF HEALTH SCIENCES:
HREC REF no. 414/2017

Project title: Developing a diagnostic tool for scapular dyskinesis

Why is this study being done?

Mr Jaco Verster, a student at the UCT Division of Biomedical Engineering, has developed a new scanning device as part of his research ***in fulfilment of the requirements for a masters' degree in Biomedical Engineering***. ***This device that can measure the position of your shoulder blade, where it sits under the skin using only a photo of your upper back. It has been established that there is a link between the position of your shoulder blade and shoulder injuries. A device that can easily measure the location of the shoulder blade will therefore be a valuable tool that doctors can use to help diagnose shoulder problems. The study is being done in order to test the scanning device on human beings to make sure that it works according to its design.***

How many people will take part in the study and how long will it last?

About 20 people will take part in the study and it will last a few weeks in total.

What do we do to decide if you are eligible to be take part?

You must be at least 18 years old in order to take part. You will be unable to joint if you had any previous surgical procedures done to your shoulder or if you have pins and needles or numbness in any part of your arms, shoulders, neck or back.

What will happen if you decide to take part in the study?

You will be asked to meet Mr Verster at either the Sport Science Institute of South Africa (SSISA) Gait laboratory or the UCT Anatomy building in Cape Town, or the High Performance Centre (HPC) at the University of Pretoria. A private room, where the scanning device will be set up, will be used for the testing and the whole test will require about 45 minutes of your time.

The testing procedure will be explained to you and you will be asked to sign in a consent form. Participation in this study is completely voluntary and you may choose to withdraw from the study and stop the session for any reason at any time without any negative consequences. Men will be required to remove their shirts, and women will be required to wear halter tops for the duration of the testing to allow the upper back to be visible. The testing procedure will follow these steps:

1. ***Your weight, height, waist circumference and hip circumference will be measured.***
2. ***You will be asked to stand in front of the scanning device with feet shoulder width apart and aligned with a mark on the floor.***

3. ***Next, you will be asked to place your arm in the first of four different positions while your shoulder blade is scanned. The scan will involve a photograph taken of the shoulder with the arm kept still for a few seconds.***
4. ***Right after the first scan, while keeping your arm still, a few small marks will be made on your shoulder using the non-permanent marker and a second photograph will be taken using the scanning device.***
5. ***You will be given a few minutes rest before moving on the next arm position. Steps 6 and 7 will be repeated for each arm position. The arm positions are as follows:***
    - ***Arm placed behind your back, with fist clenched,***
    - ***arm hanging by your side,***
    - ***arm raised 45 degrees to the side and***
    - ***arm raised 90 degrees to the side.***
6. ***This will conclude the testing and you will be welcome to leave at this stage.***

This testing procedure has been approved by the Human Research Ethics Committee of UCT (Ref nr. 414/2017).

What are the risks and discomforts of this study?

The scanning device is safe and risks associated with this study are none to minimal. In the unlikely event that any discomfort or pain is experienced due to any of the arm positions or the arm being held stationary for a few seconds, the testing procedure will immediately stop and you will be asked to withdraw from the study.

Are there any benefits to you for being in the study?

The testing done will have no direct benefit to you, however it is hoped that the result from this study would provide valuable insight that will contribute to the development the scanning device.

What will happen when the study is over?

At the end of the study, the collected data will be analysed and used to take the development of this scanning device further.

Will your test results be shared with you?

The test results will not be shared with you and will have no direct benefit for you. The results of the research will be published as a master's thesis on the University of Cape Town's faculty website.

Will you receive any reward for taking part in this study?

Unfortunately, you won't receive any rewards or compensation for travel or any other expenses.

Who will see the information which is collected about you during the study?

The information will be collected by means of photographs taken of the upper back according to the testing procedure. The photos will be taken in such a way that no one will be able to identify you on them. Information collection will be done by Mr Verster only and stored on a password protected

personal computer. The data will be used for this study only and will not be distributed to anyone else.

Who do I speak to (or contact) if I have any questions about the study?

If you have any complaints or questions about your rights and welfare as a participant in this study please contact the Principle Investigator, Dr Janine Gray, on the details provided in this consent form.

Statement of Understanding

I confirm that the exact procedure and techniques and the possible complications of the above tests have been thoroughly explained to me. I am free to withdraw from the study at any time should I choose to do so. I understand that I may ask questions at any time during the testing procedure. I know that the personal information required by the researchers and derived from the testing procedure will remain strictly confidential and will only be revealed as a number in statistical analysis. I have carefully read this form and understand the nature, purpose and procedures of this study. I agree to participate in this research project conducted by the UCT Division of Biomedical Engineering.

Name of investigator: Mr Jaco Verster

Signature: _____Date: _____

Name of Participant: _____

Signature: _____Date: _____

Name of Witness: _____

Signature: _____Date: _____

Researcher Contact Details

Principle Investigator:	**Co-Investigator:**
Dr Janine Gray Exercise Science and Sport Medicine Department of Human Biology Faculty of Health Sciences University of Cape Town, Newlands, 7725 janineg@cricket.co.za	Mr Jaco Verster Master's Student Division of Biomedical Engineering, Department of Human Biology, Faculty of Health Sciences, University of Cape Town, Observatory, 7925 vrsjac007@myuct.ac.za
UCT Research Ethics Committee Contact: Prof Marc Blockman Chairperson, Faculty of Health Sciences Research Ethics Committee Room E52.23, Old Main Building, Groote Schuur Hospital, Observatory, 7925 Tel: +27 21 406 6492	