MRes project report

B-spline surface reconstruction of the human body from 3D scanner data

By Ioannis Douros
Supervised by Prof. B. Buxton

Department of Computer Science
University College London

9th September, 1998
This report is submitted as part requirement for the Research Masters degree in Computer Vision, Image Processing, Graphics and Simulation in the Department of Computer Science at University College London.

It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed, provided the source is explicitly acknowledged.

Ioannis Douros.
Acknowledgements

I wish to express my thanks to my tutor Bernard Buxton and colleague Laura Dekker for their help and guidance in pursuing this research. I also wish to thank Elliot West, upon whose work was this project based, Pip Bull for his assistance in technical issues, and my anonymous model for assisting me when gathering raw data for this project.

Finally, I would like to thank Jonathan Wells and Nigel Fuller at the Dunn Clinical Nutrition Centre, Addenbrooke Hospital, Cambridge, for their contribution in the acquisition of volumetric data.

This work is dedicated to the memory of my grandmother, Nika Karadima-Kelemen.
Abstract

This report describes work performed at University College London as part of the Computer Vision, Image Processing, Graphics and Simulation Research Masters course. The work concerns the generation of 3D human body models from point data. Data is gathered using a rapid infrared scanner.

To avoid the complexity of general surface-from-point-set schemes, algorithms specific to body shape have been used and/or developed for all stages of data processing, from data cleaning and segmentation to final surface fitting. The work is a follow-up on the MRes project of Elliot West, student at the same course during the academic year 1996-97. Elliot West managed to devise a method of fitting B-spline curves along horizontal cross-sections of the legs and torso of the body. This project extends his work further, thus comprising another step towards the generation of a surface model that describes the whole body. The method proposed here consists of the following five subproblems: Modification of West’s method in order to make surface fitting possible for the rest of the body, resampling of the fitted curves, re-segmentation of the body so that each segment has the topology of a deformed cylinder, fitting a surface around each segment, and smooth segment joining.

The output of this method is a smooth surface, closed and continuous (with the exception of the area below the feet, which is in no way accessible by the scanner) that describes the body. The results show that this method can give a reliable body description, and that the model can also be used for calculating the body volume and surface area. There is a description of proposed methods to calculate these measurements using the model, and an attempt to compare the figures to those acquired using traditional techniques in order to provide an estimation of accuracy.

The results from this work show that further development is warranted, and also indicates possible directions in which the development should occur.
1. Summary of the problem area

1.1 Background

Recent advances in three-dimensional scanning technology have enabled the generation of high-density point data sets that describe the surfaces of real objects, including animate objects such as the human body. This means that it is now possible to produce computer-based models that describe in detail the topology and the geometry of an actual human body. These models would be welcomed in a number of application areas, particularly the clothing industry and medical research. These models would enable one to perform fast and accurate automatic measurements on the human body, in order, for example, to monitor body growth, or even design customised, well-fitting clothes.

However, such measurements are impossible to make on raw point data. A technique must be devised in order to produce a proper surface (skin), given the data. This requires an algorithm that can analyse the data and from them infer the topology (and subsequently the geometry) of the body.

![Figure 1: The surface-from-3D-point problem](image)

The Department of Computer Science at University College London has been loaned a whole body scanner manufactured by Hamamatsu Photonics [3][27], and work is in progress to exploit scanner data. The major objective of this research is a fast and robust skinning algorithm that would facilitate the development of automated body measuring tools.

1.2 Motivation

A raw point set is practically useless until it has been appropriately processed. Surface description is the minimum requirement of processing outcome, but certain applications may require further processing in order to automatically identify segments and landmark points on the body.
Clothing industry

This is currently one of the most significant areas of body modelling, since the availability of accurate body models could bring a revolution to the clothing industry, by enabling the rapid design and manufacturing of clothes that are guaranteed to fit properly. So far there isn’t really a standard way of measuring bodies and designing clothes. In spite of attempts at standardisation, each major designer has its own method of work, and there is great variation in shapes and sizes. Moreover, existing size ranges have been established during the early 1950’s, and since then the sizes of the population have changed\(^1\), which means that sizing ranges and standards need to be redefined. This requires a large scale survey on a rather huge population sample. Each member of the sample will have to be measured in detail, and doing so manually is an extremely difficult, practically impossible task. The procedure is rather time consuming, and not always very effective. The need for a new survey becomes obvious by taking a look at the statistics concerning return rates of clothing item purchases. High street retailers have a return rate of approximately 20\%, while in the case of mail order purchases, where there is no chance of trying the item before buying, that rate increases to around 40\%. This of course means loss of profit, and clothes manufacturers would be really happy if they could overcome it.

When designing clothes, the aim is to make them fit to as many people as possible. To achieve this, statistical data about the sizes of the population must be taken into account, in order to formulate sizing scales and their geometrical interpretation, so that, e.g. a 32-size pairs of trousers always have the same size and shape, even when produced by different manufacturers. Current sizes are based on surveys carried out in the early 1950’s. Since then, however, population sizes have changed, mainly due to changes in nutritional standards and general lifestyle. A new survey should be carried out, and utilisation of current scanning technology is probably the only way to measure sufficiently large population samples within reasonable time periods. Using the current traditional methods (i.e. tape measuring), it takes approximately 45 minutes to make a complete set of measurements for each body. This is an extremely long time compared to the 12 seconds required to take a scan using a Hamamatsu Body Lines Scanner, and also adds up to the fact that when using a tape there is no standard way of making the measurements. Given all these, it becomes apparent that the use of computerized techniques would be of great importance in case of a sizing survey which, according to estimations [2], should involve measuring approximately 20,000 people in order to be statistically meaningful.

Medical Applications
There are certain areas of medical research where it would be desirable to have a computerized model of the patient’s body with which to work, rather than using the actual body of the patient itself.

One such area is body growth monitoring, useful mainly in nutritional medicine for obesity and anorexia research. Acquiring successive scans of a subject at regular time intervals would make possible the detection of changes in body structure and enable study of how this can be affected by variations in types and dosages of nutritional elements (i.e. changes in diet).

Another area is that of global measurements, such as body volume and surface area. For the latter (surface area), very simply there is so far no method of measuring it, if for a living person. The only available estimates for the area of a human skin have been acquired by skinning corpses and stretching pieces of the skin on a flat surface.

As for body volume, the most dominant methods currently used are immersion tanks and pressure pods. The first method consists of fully immersing a subject into a tank of water and then measuring the volume of displaced water. The second consists of placing the subject into an airtight pod, pumping in a known quantity of air, and then inferring the body volume from the difference in pressure. Both methods are time consuming, highly intrusive, almost guaranteed to cause discomfort to the subject, and simply impossible to carry out on certain categories of patients (e.g. frail, old and unwell patients). On the other hand, using a 12-second, non-contact body scanner would enable accurate measuring of body volume and surface area while causing minimal distress to the patient.

Other Applications

Other possible areas of application for 3D body modelling include:

- **Virtual Environments**: Scanning methods can enable the automatic generation of realistic ‘avatars’ for immersive environments. In order to depict a human body realistically, a huge number of polygons is normally required, making it virtually impossible to start designing from scratch. However, owing to continuing improvements in rendering hardware and algorithms, it is now possible to handle quite detailed representations of the body, and a technique that allows users to capture the form of any human body and incorporate it into a virtual environment automatically would be very useful in an attempt to achieve accuracy and realism. An example of such models is depicted in the fairly recent trend of creating virtual characters, such as pop star Kyoko Date [23], created in Japan, and student Hori Rei [24], created here at UCL. Both indicate how body models can be used in a highly profitable

---

1 People now seem to have become bigger and fatter, and this is mainly due to the radical changes in nutritional standards and habits.
area of the media.

- **Car and aircraft interior design**: A more traditional application is the use of real body models in CAD for the purpose of ergonomics. The use of real body models in vehicles and cockpits could assist a designer in achieving maximum comfort and usability, while making the best use of all available space in case this is a crucial factor. At a later stage, one might consider introducing articulation and incorporating the mechanical dynamics of the body in order to reliably simulate experiments such as crash tests, which are currently extremely expensive and time consuming (see e.g. [25]). In the high-technology end of these industries, relevant applications become even more interesting, e.g. in motor racing, where the cockpit (and clothing) is designed around a specific driver in order to achieve maximum performance. Similar applications include clothing design for military jet pilots.

- **Robotics**: Introduction of articulation as mentioned above could also assist the study of human body mechanics towards the design of biomechanical devices, such as artificial limbs and robotic arms. So far no such device is known to be as functional and flexible as a real human arm-and-hand system. There are areas such as the manufacturing industry that would greatly benefit in case such high functionality is ever achieved.

1.3 Scanning technologies

The output of any surface reconstruction technique is strongly dependent on the original data set, which in turn depends not only on the nature and geometry of the object to be reconstructed, but also on the technology and methodology of data acquisition. In particular, when the object to be reconstructed is a human body, there are extra constraints that have to be taken into account when selecting a particular scanning method. These constraints are:

- **The size of the body**: An adult human body is very large in comparison to intended object sizes of most scanners. To make things even more complicated, a scanner optimised for scanning full grown adults is usually rather inappropriate for scanning children or abnormally small adults, because this produces an overall lower-resolution data set, and in the standard case where the scanner scans in a radial manner, this data set is also very sparse.

- **Occluded regions**: These are inevitable due to the geometry of the body, and include underarms, inner thighs, and even the area under the jaw for some people\(^2\). There are certain postures that allow minimum occlusion of such areas, but care must be taken that normal body form is not distorted by
extreme postures (e.g. postures that cause very tense muscles or skin folds that normally aren’t there when the body is relaxed and standing up).

- **Body Contact:** Data acquisition methods can generally be categorised as intrusive or non-intrusive. The former category includes all those techniques that involve direct physical contact between the sensor and the body, the latter category includes all those techniques in which subjects only touch with their feet the ground they stand upon. When the object is a human body, it is generally desirable that the method used is non-intrusive, because (a) human body is non rigid but demonstrates a certain degree of elasticity that might lead to local deformation at the point of contact with a sensor, and (b) due to the medical nature of some applications, it might not be appropriate for contact to be made with the body.

- **Stillness:** A human can remain absolutely still only momentarily. There are certain types of motion, usually unnoticed by a human eye (chest movement due to breathing, head sway, eye blinking, reflective response to environmental stimuli such as noises and lights etc.), that can nevertheless affect the quality of the scan and produce blurred, noisy datasets. It is therefore desirable that scanning time is as short as possible, ideally at most a few seconds.

All these properties described above define the type of scanner to be used. It becomes evident that rapid and accurate data acquisition has only recently become possible. The main categories of scanner technologies are described below.

**Mechanical touch probes**

The traditional way of acquiring data is doing it by hand. This is possible by using a touch probe mounted on a 3D pantograph [1]. The probe is placed point-by-point on the surface of the object, and Cartesian 3D coordinates are derived from changes in joint angle. This procedure is the worst possible for the applications in mind, since it is both invasive and time consuming. It is only good for small, inanimate, rigid objects, but completely unsuitable for human bodies.

**Shape from Stereo**

There are a few image processing techniques for inferring shape information using stereo methods (e.g. the TC\(^2\) system). Of those, the most accurate are based on photogrammetric techniques. Examples of such systems are the one proposed by J.P. Muller et. al. [31] (for modelling of heads) and also the one

\(^2\) This, however, is more stance-dependent than person-dependent, since the problem can be resolved by establishing a
by Siebert and Patterson [32]. These images undergo feature matching to find corresponding points, and shape information is derived by calculating binocular disparities. All these methods involve acquisition of multiple 2D images of the body from various viewpoints, with the cameras at known locations. However, feature point matching is not always very easy due to the low feature density of a human body. It is therefore usual to have micro grids projected onto the body in order to increase feature density.

**Photonic strip and point scanners**

Optical scanners work by projecting onto the body a stripe of light (usually visible or infrared laser), whose reflections are then measured by one or more optical sensors. Measurements are digitised and then triangulated in order to calculate the position of the point. The LASS scanner (see section 2.2 for details) and the CyberWare scanner (see http://www.cyberware.com for details) are typical examples of such scanners.

An evolved form of photonic point scanner is the Hamamatsu Body Lines Scanner, which is discussed separately and in more detail in the section that follows immediately below.

### 1.4 The Hamamatsu Body Lines Scanner

Recently Hamamatsu Photonics have developed a relatively inexpensive, but highly accurate and fast scanner based on position sensitive photon detectors [27]. One of these instruments has been installed at UCL under a loan agreement to the Department of Computer Science. With it, data may be collected in few seconds (12 sec) rather than minutes\(^3\). Data quantity is in the order of 200 Kbytes/image and accuracy is approximately 11 bits. The scanner’s main mechanism consists of 8 cameras, each one of them measuring reflections from an array of 32 infra-red LED’s, thus capturing a 256-point horizontal cross-section of the body every 5 millimetres, as the cameras slide down the 2 metres of the scanner’s working height.

There are some technical issues regarding the scanner. These issues must be addressed since they affect not only the methodology of the scanning procedure, but also the whole data processing, from segmentation to surface fitting.

Owing to the particular spectrum of light frequencies (near infra-red) utilised by the scanner’s optical system, there are some surfaces of particular colours that cannot be registered. This problem usually occurs with very dark colours, and becomes clearly apparent especially when scanning people with dark hair. Therefore, subjects are asked to wear a white swimming hat during scanning, as well as an certain posture (head tilting backwards) so that no areas are occluded.
appropriately coloured swimsuit or underwear.

![Hamamatsu Body Lines Scanner](image)

*Figure 2: The Hamamatsu Body Lines Scanner*

The other major problem is area occlusion. Despite using eight scanning heads, there are still areas of the body that cannot be accessed by any head, due to visual occlusion. Such areas are the inner thighs and the armpits. To resolve, or at least reduce this problem, an appropriate standard posture is adapted so as to maximise the percentage of visible surface area.

A detailed discussion on how the scanning procedure should be carried out, as well as a number of methodological modifications that were introduced during this project and proved to work quite well, are included later in this report.

1.5 **Scope of the project**

The aim of this work is to construct an algorithm that rapidly produces accurate ‘skinned’ body models for use in a variety of industries. Initial body data is assumed to have been produced by a Hamamatsu Body Lines scanner and processed through Elliot West’s algorithm for horizontal cross-section least squares fitted B-spline curves (a description of Elliot West’s method and its relation with this work follows later in this report). These curves (their knot and control point sequences) are known. It is

---

3 as opposed to e.g. the LASS scanner or the CyberWare scanner.
4 Throughout this dissertation, the term ‘skinning’ will be used with the meaning it is usually given by body modelling researchers, that is ‘creating a skin surface and placing it upon the body (point data)’, as opposed to the colloquial meaning of the word which is exactly the opposite (i.e. ‘removing the skin’).
also implicitly assumed that data describe a human form adopting a specific posture. The model should
cover the whole body up to a certain level of detail. This level of detail excludes detailed body parts such
as fingers and toes, as these parts present a whole new set of problems, more or less irrelevant to the
particular application areas of interest here. It is also difficult to gather sufficiently dense data from them,
owing to limitations of scanner resolution. The level of detail also excludes surface tangent discontinuities
(cusps), since they require a different way of modelling that is outside the scope of this project. In
particular, modelling of cusps would require major alterations and redesigning of the techniques applied
during the very early stages of processing. During the least-squares horizontal curve fitting, all cusp
information is lost, therefore the whole method would have to be redesigned and re-implemented in order
to detect the cusps and retain this information during and after curve fitting. This is extremely time
consuming and, in the author’s opinion, it is the subject of a separate project in itself.

The output of the algorithm to be designed should be a smooth surface that describes the body
with an accuracy of a few millimetres (required) or less (desirable)\(^5\). Furthermore, these results should be
available within a few minutes after the scan.

### 1.6 Contributions

A number of algorithms are presented that deal with the surface-from-curves problem. Previous
work on the curves-from-points has been exploited, so that their merging comprises a suggestion on how
to solve the initial problem of surface-from-points.

To handle the surface-from-curves problem, it is implied that the procedure proposed takes as
input curves (B-splines) that have been fitted to the initial point data, and uses these curves to drive a fast
and robust\(^6\) surface generation process. This is done by adequately resampling the curves, then
interpolating a NURBS surface through the samples. An example of such a technique is the one devised
by Jones et.al [5] at Loughborough University (see also section 2.2 about the LASS scanner).

Resampling the curves is not a straightforward task. The resampling strategy affects the final
surface, so a few alternative strategies are presented here, as well as a comparison between them. All of
them have been implemented and tested in order to determine which one is the most appropriate. The
conclusion drawn was that different sampling logic has to be followed for different body segments,
therefore body geometry and topology is implicitly taken into account throughout the process. Some of

---

\(^5\) Specification of this requirement derives from: (i) the accuracy of the scanner itself [27] which lies within that range, (ii)
the nature of the medical applications where lower accuracy may produce misleading results that are totally undesirable and
useless in any form of diagnosis, and (iii) the accuracy of measurements required in the clothing industry.

\(^6\) ‘Robust’ method in this context means: Accurate, insensitive to noise, outliers and missing data, and leading to a complete
surface without gaps/holes.
the alternative sampling strategies have proven to be much more robust than others, since not only do they predict and handle the sometimes bad behaviour (oscillation and self-intersection) of NURBS surfaces during interpolation, but they can also deal with the problem of poorly fitted curves on the data, since there is some sort of ‘intelligence’ included in them (they are designed in such a way that the optimum location for a sample is automatically determined, requiring no user intervention, i.e. the logic of what is defined as ‘optimum’ is incorporated into the algorithm).

Surface interpolation has been based on the standard method of B-spline interpolation through quadrilateral point grids, but this has been slightly modified in order to efficiently deal with certain topological elements of the human body, such as branching point. The result is a compound, multi-segment, yet entirely smooth surface.

The initial objective was to implement this technique for the entire body. Unfortunately, due to time constraints, implementation currently covers part of the body (torso, shoulders, legs and feet). However, the area covered includes almost all possible topologies\(^7\), even for the areas not included in the current implementation. Therefore, the current implementation can be extended fairly easily so that the surface produced can seamlessly include the rest of the body. Explanations and directions on how to do this expansion are explained later in this report.

Also, the surface skinning procedure had been initially designed to produce a method for calculating body volume and surface area. Work has been carried out to devise such a method, and a few alternatives have been considered, each one with its own advantages and disadvantages concerning speed and precision. All of them require having a surface fit for the entire body, and since this has not been implemented, volume and surface calculation are not fully implemented either. However, a theoretical presentation and comparative study of these methods is included in this work, and it would not be difficult to include any of these methods in a full body implementation, should it be necessary further to develop the proposed procedure as a whole (for details see section 8.1 - ‘Further work’).

1.7 Problem Description

In order to achieve the final objective of producing a body surface, there are a number of difficulties to be overcome. Those difficulties are listed and described in this section. All of them had to be taken into account during the work, since they constitute problems to be solved, and the aim of the project has essentially been to tackle these problems.

\(^7\) The only topology not covered so far is the top of the head and the ends of the arms (fingertips). At each one of these points, there is a hole in the surface. This hole can be made arbitrarily small but it cannot close entirely with the proposed method.
Outliers: In spite of the prior work by both Laura Dekker and Elliot West to provide algorithms that detect and remove outliers, some of them are still missed. The problem is not merely one of visualization. It is a great obstacle in correct surface reconstruction. Although outliers do not affect local measurements on well-cleaned areas of the body, even a single outlier does affect correct surface reconstruction by causing spurious phenomena (see chapter 7 for further details on the effect of outliers in surface reconstruction), which in turn does not allow correct results in global measurements, such as volume and surface area. This suggests that some effort should be put towards one of the following two directions: (i) either cleaning techniques could be revised and improved or, (ii) their shortcomings should be taken into account during the later stages of processing. Since, during this project, one of the main assumptions was to build upon existing work, the second approach was the one finally adopted (see chapters 3 and 4 for details).

Hands: The problem with hands is that scanner resolution is not high enough to provide a sufficient number of datapoints to correctly reconstruct the fingers. Furthermore, finger structure is rather complex and requires a great deal of programming effort and computational resources to model correctly. For example, each finger could present (on a small scale) the same problem as encountered at the groin/legs. A possible way to overcome this problem is to define an appropriate scanning position (such as closed fists) that hides the details of the fingers without seriously affecting the body volume or surface area.

Feet: Due to infra-red beam reflections on the Body Scanner floor, a lot of outliers are produced around the feet. Most of them are successfully detected and removed by the cleaning process, but the remaining number of correct datapoints is so small that it is almost impossible to reconstruct feet shape properly. Experiments on position and on having the subjects standing on dark-coloured blocks of known height are currently under design, which we hope will provide solutions to this problem.

Inner Thighs and Underarms: These parts of the body demonstrate the problem of missing data, since they are shadowed by other parts of the body (underarms by arms, inner thighs by legs) and are therefore not ‘visible’ from the camera heads. This, however, is not much of a problem, since the assumption of skin surface smoothness and continuity allows reliable reconstruction using B-splines.

Head: The major problem with head is sparse data. Cameras are placed circularly around the body, head surface is closer to the center than, say, that of the torso, and from each LED array beams extend radially as illustrated in figure 3, therefore their distance apart of the beams close to the head is larger than around any other part of the body, leading to sparse data (Figure 3). Given the fact that on the head there are discontinuities (lips), shadowings (ears) and fine details (eyes, nostrils), sparse data is not a pleasant surprise.
The other major problem with the head is the closed topology. One might consider the head as a deformed cylinder where all points along its upper edge reduce to one point only (summit). However, this is not convenient, as a singularity occurs while attempting to interpolate a deformed cylinder through a point grid of such topology. Another scheme must be devised, so that the head can be modelled as a smooth surface that also joins smoothly with the rest of the body. Other problems associated with the head are: Hair, that is a transient feature that occludes the real shape of the head (skull), and beam tangency around the summit, which results in very unreliable data.

- **Vertical curve fitting:** This should be done by resampling the fitted horizontal B-splines (Elliot West’s work [1]), with an equal number of samples for all B-splines of the same segment, so that columns of data points are generated, and then a B-spline is interpolated through each column. This seems to be the most straightforward way, but it will have to be tested for accuracy, as well as for the effect of outliers on the overall quality of fit.

- **Resegmentation:** Using the fitting method described just above will require a slight modification in the segmentation of the body used previously (e.g. by Laura Dekker [2] and Elliot West [1]). More
specifically, the torso should be further divided to separate the shoulder area from underarms to neck so that all segments will have the topology of a deformed cylinder. This will allow resampling as above to return a quadrilateral grid of datapoints, as required for the proposed surface reconstruction algorithm.

- **Continuity and smoothness**: The proposed surface interpolation scheme (B-splines) guarantees continuity and smoothness within a segment. However, it is also necessary to ensure continuity and smoothness between segments. A way to achieve this is to sample in such a way that, where the body topology branches, the number of samples/cross-section for the ‘stem’ (e.g. torso) will equal the sum of the number of samples/cross-section for the ‘branches’ (e.g. legs). This allows us to join columns of datapoints right across body height, resulting in a smooth vertical curve. On the other hand, it will require a large number of samples for the shoulder area (where everything sums up\(^8\)), thus forcing a trade-off between continuity and execution time.

- **Surface fitting and visualization**: After all of the above have been achieved, rectangular B-spline surface patches are defined in-between the quadrilateral grid of datapoints, in such a way that they all join smoothly and therefore define a closed, smooth, continuous surface. Rendering of this surface is expected to be a rather slow process, due to the number of patches. It will also slow things down because it means rendering a considerable number of patches. Given Body Scanner resolution compared to usual screen resolutions, fast visualization may be achieved by triangulating the orthogonal grid and rendering the triangles. This will not change the visual result (up to a certain scale, because at a large magnification it will inevitably be necessary to render the B-spline patches), which is of secondary importance anyway, and volume/area can be calculated independently (see below).

- **Volume and area calculation**: This can be done independently of visualization, using only the datapoints, the fitted B-spline curves (both horizontal and vertical), and integrals of the B-spline expression. However, it will definitely require working out the associated maths.

---

\(^8\) Given the proposed sampling scheme, we have:

\[
\text{#samples/shoulder} = \text{#samples/right arm} + \text{#samples/torso} + \text{#samples/left arm}
\]

where:

\[
\text{#samples/torso} = \text{#samples/right leg} + \text{#samples/left leg}
\]

therefore:

\[
\text{#samples/shoulder} = \text{#samples/right arm} + \text{#samples/right leg} + \text{#samples/left leg} + \text{#samples/left arm}
\]
2. Literature Review

The aim of this chapter is to present the current state of the art in human body modelling, the context of the problem, the attempts that have been made so far to solve it, and issues that still remain unsolved. Understanding of this context will provide a clearer insight to the problems that have been addressed during this project, as well as the decisions that had to be made with respect to design and methodologies.

2.1 Overview of previous work

The ‘skinning problem’ (surface from points) is not a new one. Previous attempts fall into two categories: those that require explicit information about the structure of the data points (connectivity), and those that automatically or intelligently infer the structure of the point set from the points themselves. The methods in the first category are the most generic and can be applied to any sort of object, but they do not necessarily give the best approximation to the actual object, and are also time consuming, since providing connectivity information is usually a manual task. On the other hand, methods in the second category are less generic (usually each one of them applies to a specific type of object), but they give better results because they can utilise all implicit and assumed information that is ‘hidden’ in the data.

Methods from both categories have been used in the past and the problem of body skinning has been partially tackled. A few skinning techniques from both categories are presented below. All of them have been reviewed at the early stages of the project in order to consider potential solutions for the body problem in particular. They are all recent, and most of them also well established in the areas of computer graphics and computer vision research.

2.2 LASS

The Loughborough Anthropometric Shadow Scanner (LASS) is a system specifically developed for scanning and skinning human bodies (see [4], [5] and [6] for details). It is based on a triangulation procedure. The model stands on a rotating turntable, a vertical slit of light is projected onto the body and the projection is captured by multiple television cameras. For each angle of rotation, images from cameras are combined to infer the depth of the projected slit of light, and the algorithm outputs a set of 3D points on the surface of the body, which are also partially structured as columns of data points.

A manual procedure follows. Data are segmented and key slices are extracted to provide an orthogonal grid of points. Then, a cubic spline curve is semi-automatically fitted along each slice and
column of data points. From this point on, curves are resampled at regular intervals, thus defining cubic spline patches with minimal distortion. These patches are used to build a complete surface.

There are some obvious shortcomings as a result of which it is impossible to use this method widely, e.g. for commercial purposes. The scanning procedure is quite slow (approx. 2 minutes during which the subject must keep trying to stand still) compared with the time required for the Hamamatsu Scanner, and there are several stages in the procedure that have to be carried on manually, which means that an expert has to supervise the production of every single body surface.

![Figure 4: The LASS system. (a) the scanner, (b) reconstruction of a female torso using 29 sections with 30 points per section, (c) reconstruction of a female torso using 10 sections with 12 points per section. It can be seen here that reduction of the dataset has very small impact on the final surface.](image)

However, there is a lot of potential in this method, since it avoids the complexity of other schemes, and its main core is essentially a fairly fast process\(^1\). Additionally, during its development some useful conclusions have been drawn, concerning human body modelling and representation. For example, it has been found that 32 slices of data are sufficient to satisfactorily represent a human torso if we use cubic B-spline curves and surfaces. For each of these slices, 32 control points are sufficient to represent corresponding curves, which reduce to 16 if we take advantage of lateral symmetry and store only half the body\(^2\). This means that from a dataset of 315,000 raw points we can store a recognisable torso with only 512 control points (one could see for example the Loughborough University web page at [http://www.lboro.ac.uk/departments/en/gallery/sammie/lass.jpg](http://www.lboro.ac.uk/departments/en/gallery/sammie/lass.jpg), which is a very important benefit for certain application that require fast rendering.

\(^1\) It must be noted, however, that LASS does not attempt to model the whole body, but only the torso.

\(^2\) This is an advantage for applications and communication (e.g. over the web), not really for storage itself. Disks are now so cheap compared to the cost of any body scanner, therefore the cost of storing the whole body is so utterly negligible that we would never throw away the raw data.
2.3 Hughes Hoppe

The scheme devised by Hughes Hoppe is a generic method for fitting a smooth surface comprised of B-spline patches on unstructured point data of arbitrary topology. It is extremely accurate in case of noiseless data, and also has the advantage that it requires absolutely no prior knowledge of data topology in order to generate the surface (see [13] for further details).

As one can see from the picture, Hoppe’s algorithm tends to produce very good surfaces for objects such as the one depicted. However, it is a complex procedure that demands vast computational resources. Execution time is fairly long. For the object in the picture (4000 points), skinning took 149 seconds on a 105 MHz HP735 workstation. For comparison, we recall here that a typical scan from the Hamamatsu Body Lines Scanner consists of 102500 points, which also have to go through a cleaning process to remove noise and outliers. In his thesis [13], Hoppe provides the following execution times for a number of objects represented by point data sets of various sizes:

![Stages of Hoppe's method](image)

**Figure 5**: Stages of Hoppe’s method: (a) Initial point set, approx. 4000 unorganised points, (b) initial triangular mesh estimation, (c) Voronoi partitioning of the triangular mesh, (d) Delaunay triangulation of the mesh, (e) initial quadrilateral control mesh before adaptive refinement, (f) iteratively refined control mesh, (g) final surface (the boundaries of the final patches are shown).
<table>
<thead>
<tr>
<th>Number of datapoints $n$</th>
<th>Execution time in hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.8</td>
</tr>
<tr>
<td>4102</td>
<td>1.3</td>
</tr>
<tr>
<td>10000</td>
<td>2.7</td>
</tr>
<tr>
<td>12745</td>
<td>5.2</td>
</tr>
<tr>
<td>12772</td>
<td>5.6</td>
</tr>
<tr>
<td>26103</td>
<td>7.4</td>
</tr>
<tr>
<td>30937</td>
<td>10.5</td>
</tr>
</tbody>
</table>

Plotting these figures on a time/#datapoints chart gives us the following diagram:

This diagram indicates that the complexity of Hoppe’s algorithm can be considered to be linear. Even so, this means that in order to process a cleaned body scan (approximately 20000 points, since about 80% of the points are discarded during cleaning), approximately 6 or 7 hours are required (on top of the time required for cleaning) which is not an acceptable execution time, since it is desirable to achieve an execution time of the order of a few minutes.

Such execution times are not unexpected if we take into account the steps of the algorithm, which are briefly as follows:

- **Initial mesh estimation**: An initial unoptimised dense mesh is constructed. This is done by organising data points into triplets such that triangular facets of minimum size are generated. To achieve consistency of facets, tangent planes for each point are estimated and used to construct a minimal
spanning tree. Then, contours are traced using a marching cubes algorithm.

- **Mesh partitioning and triangulation:** The number of faces is reduced by carrying out a Voronoi partitioning on the initial mesh, followed by a Delaunay triangulation.

- **Initial surface generation:** An attempt to create a first complex of B-spline patches is made using the given triangulation.

- **Adaptive surface refinement:** Number of patches is iteratively minimised by enforcing a combination of techniques including moving vertex positions to achieve better patches, solution of a least squares problem to improve vertex positions, and edge operations such as collapsing, splitting and swapping.

- **Smooth surface optimisation:** Patch topology is iteratively refined so that the mesh changes for piecewise linear to piecewise smooth. This involves computation of tangent vectors, surface subdivision and minimisation of an energy function.

It is thus obvious that Hoppe’s method is not simple at all. Furthermore, when used with scanner data (which are inherently noisy), results may appear good at a large scale, but at a higher detail level various flaws are revealed.

![Figure 6: Result of applying Hoppe’s method on a Hamamatsu Scan. Hoppe’s method has been applied after cleaning](image)

The most important shortcoming is that no assumptions on topology are made, so Hoppe’s
algorithm may infer the wrong topology. This results in phenomena such as the hands joining with the outer parts of the hips (or hands joining each other, in the case of figure 6), and inner thighs joining below the groin. Furthermore, the algorithm cannot handle missing data, so generation of a closed surface is not always the case. The result for the scanner data has many spurious holes and polygons that are generally undesirable and cause further problems. Therefore, although Hoppe’s algorithm is one of the best for general purpose surface skinning, it proves to be rather unsuitable for this particular application area of body modelling.

2.4 Stoddart and Hilton

A significant amount of work has been carried out lately at Surrey University by A. Stoddart and A. Hilton towards the development of a generic surface reconstruction technique, based on the notion of ‘deformable surfaces’ [16]. Their method, which is commonly known as ‘Slime’, generates $G^1$ continuous deformable surfaces using a generalisation of the concept of biquadratic B-splines. The surfaces produced have very good properties. They can describe any surface topology including local discontinuities (step edges, roof edges or cusps), The method has low computational complexity and can adaptively generate control points of variable density in order to describe surfaces that are very detailed at some places and very smooth and bland at others.

However, the ‘Slime’ method works on voxel data, which means that an initial set of point data has to be converted to voxel data before being submitted to any other sort of processing. This might initially sound good, as it also combines well with the idea of using a marching cubes algorithm [8] on the voxels in order to calculate enclosed volume, but in practice this is hard to do. Conversion from point to voxel form is not a straightforward task, especially with body scanner data, where noisy and missing data can affect conversion and subsequently ruin the fit. In particular missing data (armpits and inner thighs) is a problem that is very difficult to overcome using Slime surfaces at their current state of development.

2.5 Previous MRes project work

The work of Elliot West [1] during his MRes project at the UCL Computer Science department last year is the main framework upon which this project was based. Although his work has not reached the stage of actually generating a surface (due to time constraints that are inevitable in all such short term projects), it has been focused on tackling a multitude of problems that are specific to the procedure of

---

$G^1$-continuous curve segments: The direction (but not necessarily the magnitude) of the tangent vector is the same for both curve segments at the point where the segments join.
fitting surfaces to bodies given initial data from the Hamamatsu Body Lines scanner. The problems that Elliot West managed to tackle are:

- Noisy data (due to reflections)
- Missing data (due to occlusion)
- Outliers (points not belonging to body surface)
- Inconsistent data (‘ripples’ at areas covered by more than one camera)

Since the work presented here is actually a follow-up on the work of Elliot West, his work will not be presented here in detail, as it will be frequently referenced throughout this report. A full description of the steps of his algorithm can be found at section 3.5 (‘Least squares curve fitting’).

It must be emphasised here, that the work of Elliot West is the fundamental base upon which the work presented here is based. The least squares curve fitting algorithm is the solution to the problems mentioned above, and without such a solution, the method presented in this report could never be applied on body scanner data. The method suggested in this report is only feasible using the work of Elliot West as a first processing step. Use of his work allows tackling the problems with noise, outliers, missing data and segmentation, but also takes an unstructured point cloud and from that it produces structured, well defined data to work with.

2.6 Body Scanner work at UCL

Work going on at the UCL Computer Science department on body scanning is mainly carried out by Laura Dekker [2] and is focused on producing body models and measurements primarily for the clothing industry. Therefore, apart from the necessary preprocessing (noise reduction, outlier removal and segmentation), much effort has been devoted to developing an application that can output measurements on particular areas of the body, in a reasonable period of time (approx. 5 minutes). Definition of the areas where measurements have to be made is required to be automatic, which means that a series of anatomical landmarks must be located on the body (such as groin, waist, navel, nape of the neck, underbust, nipples, top of head).

The application that has been developed by Laura Dekker is currently at a stage where it can reliably detect most of those anatomical landmarks on most common body types (i.e. not for children, obese adults, and the aged) and also produce all relevant common measurements used in the clothing
industry. This is preceded by an extended preprocessing operation that removes noise and outliers and also segments the body.

Figure 7: Examples from Laura Dekker’s work. Data has been acquired using the Hamamatsu Body Lines Scanner. The dataset has been cleaned and segmented (each segment shown here in a different colour). Landmark points along the torso are also located, and a quadmesh surface has been fitted around the points.

It must be noted, that execution speed is remarkably high, as it only takes a couple of minutes on a 90 MHz Intel Pentium machine to clean the data and carry out the processing required to clean the data, segment the body and locate the landmarks for a whole body. The application also attempts to fit some surface on the points using a simple quadmesh algorithm.

2.7 Other body modelling work in progress at UCL

Apart from this work, there are also a number of people within the UCL Computer Science department that are involved in body modelling in other ways. The VR/graphics group is involved in the COVEN (‘Collaborative Virtual Environments’) project [28]. They need virtual humans to populate their virtual worlds and also accurate, realistic models of real people so that users of a virtual environment can also be represented in the virtual world. Examples include applications such as virtual meetings, e.g. for design, and the creation of virtual studios and advanced forms of interactive TV. For other work on 'body centred interaction' they need virtual humans to represent basic actions such as walking or running.
Anthony Steed is a research fellow, working on the COVEN project. He is interested in a harness to put the body into DIVE\textsuperscript{4} [29], and in the proposed standard VRML representation for humanoids [30]. Gideon Amos has just completed an MRes project on positioning human body models, for automotive applications. He is now starting a PhD on human body modelling, with an emphasis on human movement and animation. David King, currently an MRes student, is involved into the development of Hori Rei, the virtual student currently under development at UCL\textsuperscript{5}.

\textsuperscript{4} DIVE (Distributed Interactive Virtual Environment) is an internet-based multi-user VR system where participants navigate in 3D space and see, meet and interact with other users and applications. The DIVE software is a research prototype but tends to become a sort of standard platform among VR researchers.

\textsuperscript{5} For details see: http://www.cs.ucl.ac.uk/students/dhking
3. Design and Methodology

In this chapter, the proposed algorithm as a whole is presented, as well as the concepts that have driven the design of the whole technique. There is a description of the algorithm steps, and an explanation of basic terms that are going to be frequently used from here on. There is also a presentation of the main framework upon which the whole work is based, and a number of explanations regarding the design and methodology decisions that had to be made with respect to that framework.

3.1 Overview of the proposed algorithm

The algorithm presented here is the one finally implemented, so it does not include surface generation for the arms and head, but only for torso, legs and feet. The steps of the algorithm as a whole are the following:

- Fit horizontal LS B-spline curves along the data slices
- Detect key slices (feet, groin, underarms, top of head)
- Resample the groin slice at uniform intervals
- Re-order and calibrate the groin-slice sample sequence
- Produce torso samples by growing upwards
  (use the cord bisector growing scheme)
- Refine the mid-bust sample column
- Produce leg samples by growing downwards
  (use cord bisector OR bounding box scheme)
- Divide the sample set into secondary subsegments
- Fit a B-spline surface on each subsegment
- Smoothly join the B-spline surfaces

Detailed explanation of each step is presented in the sections that follow. However, to give the reader a better understanding of the algorithm as a whole (which is necessary in order better to understand the sections that follow), a very brief description of some terms encountered in the box above is given here:

- **Slices** are groups of co-planar data points that lie on a horizontal plane and represent horizontal cross sections of the body. It is easy to group the data in slices, because this is the structure in which they...
are given by the scanner.  
- **Key slices** are slices that lie at the same level as certain landmark points of the body that are important for the purposes of surface reconstruction. For example, *groin slice* is the set of points that lie at the same horizontal plane as the branching point of the groin.
- **Sample sequence** is a sequence of points acquired by evaluating the parametric expression of a B-spline that has been fitted along a slice, for various non-decreasing values of the parameter.
- **Mid bust** is defined as the area on the front side of the torso and around the axis of lateral symmetry, i.e. the area that starts from the sternum, goes all the way down the middle abdomen, through the navel and ends at the pubic area.
- **Segment** (or primary segment) is any group of data points or sample points (the definition of which are the semantically feasible segments for the purposes of the method proposed here follows later in this chapter).
- **Subsegment** (or secondary segment) is any subset of a segment (the definition of which are the semantically feasible subsegments for the purposes of the method proposed here follows later in this chapter).

### 3.2 Problems

The main problems that were encountered during the design of the proposed system and had to be taken into account were the following:

- **Very poor quality of fit for some of the least squares B-spline curves.** While the least squares B-splines algorithm as given by West [1] will generally produce very well-fitted curves for most of the body, there are some slices for which quality of fit seems to be systematically bad. Areas where the problem becomes more serious are those close to the underarm and the lower legs. In the first case (underarms), the algorithm tends to produce self intersecting curves. These curves have the very interesting property that there are some parts of them such that their union adequately describes the correct shape of the slice, and the rest of the curve is just a series of spurious loops running away from the real data. This problem is tackled by applying an adaptive sampling strategy that only takes samples from the valid areas of the curve (we **do** know which areas are valid - or at least we can assume pretty safely - based on the assumption that there has to be a certain degree of distance continuity between successive slices. Further

---

1 In the initial dataset, points on a slice are not exactly co-planar. They have a helical geometry because each LED in a head is fired sequentially while the heads move downwards. However, the height difference is negligible and in practice we substitute the height of each point in the slice with the mean slice height.
details on this are included in sections 3.3 and 4.2 through 4.4). In the second case (lower legs), the ends of the curve intersect each other and extend beyond the surface of the leg, in most cases folding inwards, while the rest of the curve is valid. Again, the sampling method attempts to overcome this by incorporating some degree of ‘intelligence’ to discard invalid curve segments.

- **Topological irregularities** of the least squares B-splines. The curves fitted along the slices are closed, but the closing point where they start and end can be in any direction (front, back, left, right) with respect to the slice centroid. Moreover, each one of them may run either clockwise or counter clockwise as the parameter value increases, and information on this is not explicitly recorded anywhere, so it must somehow be inferred. Otherwise, the samples might be geometrically valid, but their connectivity structure will be incorrect. This problem is overcome either by analysing the sample sequence and correcting its structure, or by applying a sampling strategy that by definition produces samples in the correct order (both cases are described later in the next chapter).

- **Semantic differences** for the B-splines between the curve fitting algorithm and the surface fitting algorithm. The curve fitting algorithm [1] uses absolutely uniform knot sequences. The current surface interpolation method is focused on end-point interpolation and uses knot sequences where the values of the three knots at each end are identical (to ensure that the curve starts at the first data point and ends at the last), and the rest of the values range uniformly between the end values. During implementation, extra care had to be taken so that no errors would occur during transition from one semantics to another, so that the same curve forms would result even when using different knot sequences.

- **Poor behaviour of B-spline surfaces for helical and unevenly distributed data.** When interpolating a B-spline surface through a quadrilateral grid of points, some conditions have to be met for the surface to be acceptable for the particular application domain. In case where there are big differences in the distances between neighbouring data points, so that some of them are very close to each other and some others too far apart, unexpected concavities will emerge in the dense areas of the grid that do not correspond to the geometry of the human body surface. Furthermore, if data are helical, i.e. each row of data is ‘dislocated’ with respect to the neighbouring rows (such that columns of rows demonstrate a ‘zigzag’ type of geometry), the resulting area has ripples and is not the smoothest possible. Much of the design effort has been devoted to avoiding these ripples. The problem is illustrated and analysed in detail later in the next chapter.

- **Conical type singularities.** Quadrilateral B-spline surfaces are very good for representing objects that have a deformed cylinder topology, i.e. one edge coincides with its opposite and there are two open
ends. In the degenerate case that one of the ends is closed (all points belonging to the row of data at one end of the grid coincide) the object becomes a deformed cone and interpolating a quadrilateral surface on it is a problem. That is because interpolation involves a linear system of equations, where the matrix of coefficients becomes singular, so that the system has no unique solution. This topology is encountered at the top of the head and at the tips of the fingers. In the implementation provided, these parts of the body are not included, therefore the problem remains unsolved and will inevitably be encountered in attempts to further develop the proposed algorithm. The proposed algorithm was tested with phantom data and proved to be inadequate indeed. A suggestion for the solution of this problem is presented in chapter 8.

3.3 Assumptions

The surface reconstruction technique proposed is not a generic. For the whole technique to work, a number of assumptions are made. These assumptions have determined both the design and the implementation, and they are present throughout both the theoretical model presented in this report and the implementation code provided. They are therefore presented here and must be kept in mind throughout. Most of the design decisions have been based on these assumptions:

- The object whose surface is to be reconstructed is a human body.
- The body is complete (no mutilated parts), and belongs to an adult.
- There are no serious deformities due to disease (e.g. tumours, big cysts or blisters etc.).
- The data set is provided by a Hamamatsu Body Lines scanner, and has all the properties (density, size, structure) defined by the specifications of that Scanner [27].
- The subject (human) is standing upright with feet either directly touching the scanner floor or standing on a couple of rectangular, dark coloured blocks of equal and known height, length and width.
- The length and width of those blocks are such that any adult human can stand on them without any part of the foot protruding beyond that length and width.
- The height of the subject is unknown.
- The sum of: (a) the height of the subject, and (b) the height of the standing blocks - if any - does not exceed 2 metres.
- The subject is wearing light coloured underwear that is neither too baggy, nor too tight, so that the actual form of the body (as if the subject were completely naked) is subject to a minimal distortion, ideally

---

2 Quadrilateral: We refer here to surfaces having the topology of a deformed rectangle. More details on quadrilateral surfaces are included in chapter 5, where the surface interpolation method is explained.
of the order of less than a couple of millimetres.

- The subject is wearing a light coloured swimming hat such that hairstyle features are reduced as much as possible and the shape of the scalp is preserved to a reasonable level.
- There is a distance between the feet such that there is a branching at the groin downwards, and the legs do not touch each other at any point below that branching.
- Arms are positioned in such a way that they are pointing downwards, so that there is a branching located at each armpit. The arms/hands do not touch each other or any other part of the body at any point beyond that branching.
- Both armpits are located at the same horizontal level (subject is set to such a posture that this constraint is met).
- The ends of both arms/hands are located at the same horizontal level which is higher than the level of the groin (again, subject is set to such a posture that this constraint is met).
- There is no other information regarding body geometry, sizes, measurements and analogies.

3.4 Scanning procedure and data clipping

Scanning the subject is not a trivial task. Operating the scanner is very easy, but having in mind that the final aim of the scan is a properly interpolated surface, extra care must be taken in order to do the correct preparation, and especially finding the optimal posture that provides minimum occlusion as well as the richest dataset possible.

Since we want to capture the form of the body, subjects obviously have to be scanned without their clothes on. In practice, most subjects are scanned in their underwear and not completely naked. Owing to the infra-red spectrum of light used by the scanner, very dark colours cannot be captured. Therefore, light-coloured underwear should be used. It should be form-fit, without dangling pieces of fabric, or other decorative elements that lie beyond the natural surface of the body. However, it must not be exceedingly tight, or the body form will be altered as various parts are squeezed out of shape.

For similar reasons, it is best if subjects wear a light-coloured swimming hat. Hairstyle is a transient body feature, and for body modelling purposes the shape of the skull and neck is what is really of interest. In addition to that, very dark hair might cause missing data problems, as it may not always be captured by the cameras.

As for the posture, there has been some debate [14] as to which posture is optimal, and how ‘optimal’ is defined in this context. Apparently, we should seek the posture that minimises noise and occlusion and produces a rich dataset. Posture must be implicitly known throughout the skinning process,
as it generates a series of topological assumptions to assist fitting [2]. The points of interest here are the distance between the legs, the position of the arms (forwards, upwards, downwards), and the distance between arms and torso.

Arm position is generally accepted to be such that arms are pointing downwards, so that there is no shoulder distortion and arm muscles are loose and relaxed. It is also much easier to hold the arms still in that position. The problem is to find an optimal distance between arms and torso. There is a limit imposed by the scanner spatial range that does not allow the arms to be very far from the torso (more than 30cm). It would be desirable to keep them as much apart from the torso as possible to guarantee proper segmentation. The problem becomes even more apparent in the armpit area, where arm and torso touch even below the branching point, especially in the more obese subjects. During the scanning sessions that occurred for the purposes of the research, several variations on arm postures were tested. The one that proved to have the best behaviour is with the upper arms at an angle of approximately 40° from the torso, and elbows bent so that lower arms are approximately vertical.

![Figure 8: Illustration of arm position. (a) bad, (b) good, (c) better](image)

Hand position was another problem to be solved. Since the scanner resolution is not sufficient to capture all fingers separately (and modelling them separately is beyond the scope of this project), a position had to be adopted such that all fingers join together. The initial thought was to have all fingers closed in a clenched fist, but this proved to be a rather bad idea. The palm reduces to an occluded cavity, and first-order digits (figure 9) form an approximately horizontal surface, almost tangential to the scanner beams, and produce a lot of outliers.
Finally, the most appropriate position was found to be the one with fingers straight downright and touching each other, and thumbs bent so that their tips do not stick out of the bulk of the hand. This makes virtually all of the surface accessible to the scanning heads, and also reduces singularities (tips) to only one per hand (the tips of the middle fingers).

Legs and feet were another problem to be solved. First, the groin area demonstrates the same problems as the underarms. Legs touch each other below the crotch, so it is desirable to keep the feet far apart, yet not too much, otherwise the shape of the pelvis will be distorted. Given the particular subject used (medium build), a distance of 30cm proved sufficient, but this will have to be greater for bulkier subjects.
Feet used to demonstrate another serious problem. Owing to their proximity to the scanner floor, the area around them is full of outliers, because reflection of the beams by the scanner floor. These reflection outliers are usually of high intensity, so the cleaning algorithms commonly used cannot detect them as outliers.

The idea suggested here was to have the subject stand on a couple of dark coloured rectangular blocks of known height (supplied courtesy of Benjamin Buxton and Charlie Buxton). There is a double advantage to this. First, dark colour eliminated reflections, and second, even if there were outliers around the blocks, it would be very easy to get rid of them since we know that whatever lies within the height range of the block does not belong to the body so they can be removed by clipping.

![Figure 11: Noise removal around feet after data cleaning: (a) without the blocks (b) with the blocks](image)

This idea was tested by scanning a subject using blocks of various heights, and it proved to be even more efficient than expected, as these reflections were almost totally eliminated. Best performance was achieved when using blocks with a height of 8cm.

### 3.5 Least squares curve fitting (overview)

For the fitting of the curves along horizontal cross-sections (slices) of the body, the method proposed by Elliot West [1] is used. The algorithm consists of two stages, data preprocessing and the actual fitting stage.

During the first stage, data are prepared in order to acquire slices of valid data points, so that a curve can be fitted around each one of them. This stage includes cleaning and segmentation. The steps of this stage are the following [1]:
• **Data cleaning:** Each scan produces a total of 102,500 points, many of which are noisy and inaccurate, and therefore have to be removed from the dataset. There are six types of such points, each one handled in a different way. These types are: (a) low intensity false points, removed by intensity thresholding, (b) scanner structure (internal walls) points, removed by spatial thresholding since their locations are known, (c) points from first/last LED of each camera that are ignored because they fall on the body almost tangentially, (d) high intensity outliers, removed by using a nearest neighbour method, (e) reflections, removed by checking whether a point is in the same quadrant as the cameras that detected it, and (f) inconsequential slices above the head, removed using a thresholding on the number of valid points that have remained after all other cleaning procedures.

• **Body growing and segmenting:** A number of techniques are applied to detect which points belong to which part of the body. This is done using a growing procedure that starts from the slice located at 15% of the body height (guaranteed to be legs) and keeps detecting leg points upwards until the groin is detected. Then, it continues growing up the torso until underarms are detected, and body growing continues downwards to the arms and upwards to shoulders and head, until all valid points are marked as leg points, arm points or torso points. Local criteria of proximity and topology are applied to determine groin and underarm location.

• **Data association:** In each slice of data there are now a number of valid points. Before a curve can be fitted around them, their order must be determined. For simple slices (arms, legs) this is done by calculating the centroid and then ordering the points by angle (each arm and leg is segmented separately). For complex slices, a neighbourhood scheme is followed starting from a seed point.

The steps of the second fitting stage are for each slice the following:

• A knot sequence is generated, and for each data point a parameter value \(w\) is determined. The objective is to minimise the error distance \(\|p_i - S(w_i)\|\), where \(p_i\) is the i-th data point and \(S(w_i)\) is the point spawned by the tensor product for the value of parameter \(w\). Error minimisation is done by iteratively moving the control points (this is a least-squares problem which is solved using Cholesky decomposition)

• Control points are updated (adjusted) so as to produce a smaller error value, and the process is iteratively repeated until total error drops below a certain threshold or until the number of iterations
exceeds a maximum limit.

The result of this method is a set of cubic B-splines that are closed around horizontal slices of the body. For each curve, all control points are coplanar and the distance between successive slices is 5mm (this is the ‘sampling frequency’ of the scanner). No other assumption can be made about the curves. The starting point is unknown (it can be either on the front or in the rear or in any other direction of each segment) and can be different for each slice. The orientation of each curve (clockwise or counterclockwise) is also unknown and can be different for each slice. This lack of explicit knowledge about the starting point and the orientation is not a problem for the surface skinning though, since this, as it will be shown later, is tackled by following an appropriate sampling strategy.

3.6 Body Resegmentation

The output of the horizontal curve fitting method is a set of curves structured in such away that it is very easy to detect the boundaries of the segments. The output structure is very convenient for the surface skinning technique developed during this project. The number of curves per slice can easily reveal which part of the body the slice is located at. The scanning posture proposed at section 3.4 guarantees that the tips of the fingers are at a level higher than that of the groin\(^4\), so if we keep that particular posture in mind, we have:

- From floor to groin: 2 curves/slice
- From groin to finger tips: 1 curve/slice
- From finger tips to armpits: 3 curves/slice
- From armpits to top of head: 1 curve/slice

This sort of segmentation is convenient for the skinning scheme that follows, because all segments produced have the topology of a deformed cylinder as desired\(^5\). We therefore have the following segments:

- Left leg

---

3 In the implementation given, the head is actually considered to be included with the torso.
4 Here defined geometrically (as the point where the legs no longer touch each other), not anatomically.
5 The head is actually a deformed cone, but for the moment we can regard it as a cylinder if we omit the top slices and leave the summit open. More details on that can be found in chapter 8.
- Right leg
- Torso
- Left arm
- Right arm
- Shoulders/head (one segment)

From now on, these segments will be called ‘primary segments’

Figure 12: Segments as inferred from the output of the horizontal curve fitting procedure

Apart from this segmentation, and for the purposes of skinning, some further segmentation has to take place, such that each primary segment is divided into secondary segments. Such a segmentation is carried out specifically to make surface generation easier, and is defined so that secondary segments abide by the following constraints:

- If we regard each secondary segment as a quadrilateral, we can define ‘left’, ‘right’, ‘top’ and ‘bottom’ edges. A segment is a deformed cylinder if its left and right edges coincide (e.g. a leg) but its top and bottom edges are not related to each other.
- Each primary segment is located on top of another one, in the sense that e.g. the ‘top edge’ of the leg
joins with the ‘bottom edge’ of the torso (i.e. we have torso ‘on top of’ legs, shoulders ‘on top of’ torso and arms.

- The left (right) edge of a secondary segment can only join with the right (left) edge of a secondary segment that belongs to the same primary segment.

- If a primary segment joins with another primary segment and there is no branching, they must have the same number of secondary segments, which must also join in a one-to-one manner.

- If a primary segment (‘stem’) joins with another two primary segments (‘branches’) in a branching manner, then the number of the secondary segments of the ‘stem’ must be equal to the sum of the numbers of secondary segments of the ‘branches’.

- When the top (bottom) edge of a secondary segment joins with the bottom (top) edge of a secondary segment that belongs to a different primary segment, then their left (right) edges must join smoothly to produce one smooth curve.

If these requirements are met, then the whole body surface can be divided into quadrilateral surfaces that follow an orthogonal connectivity, even if the body has three branches. Figure 13 shows how to perform such a segmentation on an object with a single branching.

*Figure 13: 2-level segmentation for a simple branching object (e.g. a torso - stem with two legs - branches). The primary segments are three: the torso and the two legs however, the torso is further divided to two secondary segments. One can easily check that this segmentation meets the constraints mentioned above.

One can see that joining the segments smoothly is now conceptually easy. Figure 14 shows how these ideas can be expanded to the rest of the body, and therefore indicates how the body should be
topologically segmented.

Figure 14: 2-level segmentation for a whole body. Solid curves show boundaries of primary segments. Dashed curves show boundaries of secondary segments.
4. Curve resampling

When the least-square B-spline curves (horizontal slices) have been acquired, resampling is necessary in order to produce a set of points through which the surface will be interpolated. In the sections that follow, a number of curve sampling strategies are discussed, and the advantages and disadvantages of each one are presented. These strategies are all compared to each other in terms of performance (they have all been implemented and tested) in order to extract conclusions as to which one is the best. The conclusion drawn was that, since each part of the body has its own topological characteristics, a different policy must be used for each body segment. Sampling methods are here presented in the order that they were devised and tested, since each one was designed in an attempt to resolve the shortcomings of the previous one.

4.1 Uniform parametric intervals

The most straightforward idea was to sample each curve (slice) by evaluating it at equidistant values of the parametric space that defined it. This has the advantage that samples are more dense at parts of the curve where detail is higher, and sparser otherwise. Implementation of this sort of resampling is completely straightforward.

However, samples require further processing in order to produce a proper dataset. For example, we need all sample sequences for the torso slices to begin at the spine area and continue clockwise. With the uniform interval sampling scheme, the sequence will begin where the curve begins (on this, no assumptions can be made, the curve can start from anywhere) and the rotation direction will be the same the curve direction (again, no assumptions can be made on the direction).

Therefore, after a sample sequence is acquired, it has to be re-ordered and calibrated. The way to do that is illustrated in figure 15.

First, the sampled that has to be defined as the first in the sequence has to be detected. For this purpose, the centroid of the samples is calculated. This centroid defines a x-axis of symmetry (front to back) and a y-axis (left to right)\(^6\).

If we want the sample sequence to begin near the spine area, the sample to be first has to:

- be the one closest to the y-axis
- be on the positive size of the x-axis

\(^6\) These axes are parallel to the respective x and y axes of the coordinate system. Only their point of intersection depends on the shape of the slice, not their direction (i.e. no PCA is used, and no second order moments are calculated). Their direction is the same for every slice on the body.
Figure 15: Re-ordering and calibration of samples: In this slice (top view), point A is the centroid. Point B is the one chosen to be the first in the sequence. Point C is the fourth. If the determinant of triangle ABC is negative (not the case for this picture) then the sample sequence requires inversion so that direction is clockwise.

Given these constraints, detection of the first sample is now straightforward. For all sample points whose value of x-coordinate is greater than the x-coordinate of the centroid, we calculate the distance from the y-axis (that is the absolute value of the difference between the y-coordinate of the sample and the y-coordinate of the centroid). The sample for which this distance is the smallest, is defined to be the new first sample, and the whole sample sequence is accordingly shifted cyclically.

To calibrate the sequence in terms of rotational direction, we need to detect the direction of the sample sequence. To do so, we need three points: The centroid, the first sample, and another sample after the first (in practice if we have about 15 samples/slice then the third or fourth sample appears to work well for this purpose).

These three samples define a triangle whose direction is the same as the direction of the sequence, so all that needs to be done is to find out the direction of the triangle, which is a much easier task. This is done by forming a determinant with the coordinates of these three points and calculating it.

If the determinant is positive, the sequence is clockwise and no calibration is needed.

If the determinant is negative, the sequence is counter-clockwise and needs calibration, i.e. the order of the samples is reversed.

If the determinant is zero, then the three points selected are collinear and no conclusion can be drawn. In this case, if the third point chosen was the i-th in the sequence then we choose the (i+1)th instead and repeat the calculation.
This method seems to work quite well for some parts of the body, but produces a poor skinning result in problematic areas of the body where least-square splines are poorly fitted due to missing data (e.g. legs). Also, since there is no functional dependency between samples in successive slices, there might still be a helical connectivity between them (see figure 16) which results in ripples in the final surface.

![Figure 16: The problem caused by helical geometry between successive slices. In this leg segment one can see how control points are helically distributed just above the ankle, thus causing a rippled surface.](image)

This problem indicates that, in order to get a proper surface, another sampling scheme has to be followed, where the sample on one slice depends on the location of the respective sample on the previous slice (growing). The methods that follow utilise this sort of logic. In the final implementation, uniform sampling is only done on the slice sampled first in order to produce a ‘seed sequence’ from which samples on the rest of the slices are grown using some other method.
4.2 Torso slice growing based on closest distance from previous sample

In order to get a set of samples such that no helical structure emerges, one of the simplest ways is to compare with the samples of the previous slice. The i-th sample on the current slice is defined as the point on the current slice with the minimum distance from the i-th sample on the previous slice, as illustrated in figure 17.

![Figure 17: Slice growing based on closest distance.](image)

This method also handles the case of very poorly fit curves, such as the ones shown in figure 18, which are a typical case of poor fit using the least-squares algorithm. In this case, there are parts of the curve that represent the correct form of the slice, as well as spurious loops that represent noise. Using the closest distance method, samples are not taken on these spurious loops, since the distance between points that belong to the loops and samples from the previous slice is generally larger than the distance between points that belong to the well-fit parts of the curve. Hence, if the seed slice is well-fitted, sampling is fault tolerant even if the rest of the slices are contaminated with spurious loops. This was a problem that could not be solved using the uniform sampling scheme.

![Figure 18: Fault tolerance of the closest distance sampling scheme. The current slice (dotted line) is very poorly fit, and some of its regions do not represent body form. However, samples are chosen so that they have the closest distances from the previous slice (solid line), therefore the distances closest to the well fitted regions of the curve, as the points of the latter tend to be closer to the real surface of the body. This helps to get rid of phenomena (such as the spurious loop in the top right part of the picture) that arise during curve fitting](image)
The major disadvantage of this method is that there is no constraint to guarantee a fairly uniform angular distribution of samples. As a result, it sometimes happens that neighbouring columns of samples (a column here is defined as the ordered set of i-th samples from each slice) tend to converge as we move away from the seed slice. This, given the usual behaviour of B-spline surfaces, means that there are some slices with a very uneven distribution of samples, which produces vertical folds and ripples in the final surface. This convergence is mainly observed where there are concavities in the slices, such as the groin and sternum areas. In the following section, a modification of this method that attempts to overcome the convergence problem is described.

### 4.3 Torso slice growing based on intersection with curve normal

The idea here is to calculate the normal to the curve in the previous slice at the i-th sample, find where it intersects the current slice, and set the point of intersection to be the i-th sample on the current slice. If there is more than one such point of intersection, the one that is closest to the i-th sample on the previous slice is chosen. The logic of this scheme is illustrated in figure 19. Normal calculation is fairly easy to do analytically on a B-spline curve. It requires calculating the tangent vector and then finding the normal to the tangent vector. Calculating the tangent vector on a B-spline is also not difficult, since it only requires differentiation of the parametric form of the curve (trivial, because it is of polynomial form), which in turn can be worked out from the associated DeCasteljau triangle.

![Figure 19: Slice growing based on intersection with curve normal.](image)

The shortcoming of this method is that for certain areas of the curve where curvature is high, the direction of the normal might be such that:

- Either intersection is at a point such that the proper sequence is overridden (figure 20, top), or
- There is no intersection point (figure 20, bottom)
Figure 20: Problems of slice growing based on intersection with curve normal.

A further refinement of the resampling scheme that attempts to overcome this problem is presented in the following section.

4.4 Torso slice growing based on intersection of cord bisector

The sampling method based on the analytical normal is problematic because it forces the samples to grow with respect to the curve, and not with respect to the sample sequence itself. This is why consistency of sampling distribution cannot be guaranteed. The modification presented in this section takes sample distribution into account, and is the one used in the final implementation for the torso area, combined with a minor emendation described in the next section.

The idea here is that, in order to get the i-th sample on the current slice, instead of intersecting the slice with the normal at the i-th sample to the previous slice, we intersect with the bisector of the line segment defined by the \((i-1)\)th and the \((i+1)\)th sample on the previous slice. This is illustrated in figure 21.
This way, the likelihood of overlapping in the ordering of samples is significantly reduced (given the particular kinds of curves for this application). It has also been empirically observed that the frequency of “no intersection” is much lower. Using the two schemes on the same number of scans (nine scans), there was an average 0.3% rate of “no intersections” using the analytical normal scheme, whereas the cord bisector scheme always gave intersections.

Fault tolerance is even higher, because this scheme is not affected by spurious cusps on poorly fit curves that would normally divert the analytical normal.

4.5 Mid-bust refining

The scheme described above works very well for the torso area, with the exception of the sternum, where the concavity in the horizontal curves (especially for female subjects) may cause sample column convergence as in the case of parametrically uniform sampling. However, this convergence is systematic and can easily be tackled as follows:

Given that sample sequences start on (or near) the spine and that there are N samples/slice, the column of samples that needs to be corrected is the i-th column, where i is such that 2i = N, i.e. the sample column that is in the middle of the sequence\(^7\), and corresponds to the area in the middle of the front side of the torso (from the crotch up to the sternum). These samples have to be corrected, after the intersection based sampling has been completed. Then for each slice, the i-th sample is redefined as the point whose x-coordinate (the one that runs from left to right) is equal to the mean of x-coordinates of the (i-1)th and the (i+1)th sample on the previous slices (figure 22). This guarantees that the i-th column

\(^7\)This is true if N is even, but since N is always even in the case of the torso. This is because N=a+b where a is the number of samples/slice for the left leg and b is the number of samples/slice for the right leg. Since a and b are in practice equal, they are either both odd, or both even, so N is always even.
of samples is always halfway between its neighbouring columns, therefore it can never converge to any of them, thus producing a fairly uniform sample distribution in the area.

![Diagram showing mid-bust refining](image)

**Figure 22: Mid-bust refining. Mid-bust sample is set so that \( d1 = d2 \).**

With this modification, the sample growing scheme based on cord bisector provides very good results for the torso area (see also: chapter 7 - results).

### 4.6 Leg slice growing based on bounding box

The geometry of the legs is slightly different, and the cord bisector scheme does not work very well. The problem is that, as we grow from the groin slice (seed slice) and downwards, sample column convergence begins at the knees level, due to the concavities on the sides of the knee-caps. As growing moves further down, samples on the front side of the leg tend to diverge sideways, and finally the shape of the foot is lost (Figure 23).

![Diagram showing loss of shape for foot slices](image)

**Figure 23: Loss of shape for foot slices.**
For this reason, an alternative method has to be used for the legs. The suggestion is to force sampling all around the circumference of the leg slice by calculating the bounding box of the slice and setting sample points wherever the bounding box is tangent to the slice. These samples drive the rest of the sampling procedure, e.g. from them the bisectors are again extracted and drive an intersection process (in the case of 8 samples per slice) as illustrated in figure 24.

![Diagram of leg slice growing based on bounding box](image)

**Figure 24:** Leg slice growing based on bounding box. Primary samples are those defined by the bounding box during a first pass. In the second pass, secondary samples are generated (in this picture using the cord bisector method) using the primary samples.

The major problem of this method is that although it guarantees that samples will be taken from all around the slice, there is no sense of ‘growing’ so again helical topology might emerge. In addition, the method is hard to extend for the case of more than 8 samples per slice, which is a restriction that affects the numbers of samples for the torso and reduces the flexibility of the whole algorithm.

### 4.7 Leg slice growing based on lateral plane intersection

The final idea for a solution to the problems of legs resampling was to design a scheme that involves a plane that rotates around the main axis of the leg, and samples are taken where the plane intersects the curves. This seems to be an optimum solution for the legs, since it guarantees no helical phenomena, and also a uniform angular distribution of samples around each slice. The geometry of the legs (roughly circular slices, no serious concavities) seems appropriate for this method, as opposed to the torso. However, due to time constraints this method was not implemented, so no results or conclusions on it are presented in this report. In case of further development, some further details on this method are given in section 8.1 (‘Further work’).

---

8 i.e. There is no dependency/correlation between samples of one slice and samples of the next slice
5. Surface fitting

This chapter contains the description of the procedure that generates the surface itself. This procedure is here described stepwise, starting from a fairly simple idea that is expanded to evolve to a method that covers the whole body. The problem breaks down to fitting a B-spline curve through a sequence of points. Once this is achieved, the method can be expanded to generate surfaces that interpolate through a point grid of orthogonal topology. Then, surface interpolation can be applied to point data that describe the body, provided that these data are segmented in such a way that they comprise a set of orthogonal point grids (whose union gives the body surface as a result).

5.1 B-spline curve interpolation

The problem of B-spline curve interpolation [26] can be formulated as follows: Suppose we have an ordered sequence of \( n \) data points: \( p_1, p_2, \ldots, p_n \). What we want is to generate a sequence of B-spline control points \( v_i \), such that the B-spline curve starts at \( p_1 \), goes through all data points in the order given, and ends at \( p_n \). Here we consider the case of cubic B-splines, but the method can be applied to B-splines of any degree. However, the implementation presented is for 3rd degree (cubic) curves, which are the standard in computer graphics. Moreover, cubic curves are the most convenient to use in this particular application domain, because the interpolation problem for the cubic case has exactly the number of degrees of freedom that is desired (explanation of what this means follows later in this section).

Let \( F \) be the curve in question, where \( F(t_i) = p_i, i = 1, 2, \ldots, n \) and \( t_i \) is a given sequence of knot values. We suppose that the blossom \( f \) [26] corresponding to \( F \) is the three parameter polar form \( f(u_1, u_2, u_3) \).

To start solving this problem, we first look at the knot sequence required. We know that we want endpoint interpolation (i.e. curve starts at \( p_1 \) and ends at \( p_n \)) so:

- The first three knot values must be identical and equal to \( t_1 \), thus giving \( p_1 = f(t_1, t_1, t_1) \).
- The last three knot values must be identical and equal to \( t_n \), thus giving \( p_n = f(t_n, t_n, t_n) \).

This means that the knot sequence must be of the form: \( t_1, t_1, t_1, t_2, t_3, \ldots, t_{n-2}, t_{n-1}, t_n, t_n, t_n \). This gives a total number of \( n + 4 \) knots, which means we are going to have a total of \( n + 2 \) control points.
(because the curve is cubic). We use the notation \( v_{-2}, \ldots, v_{n-1} \) (instead of \( v_0, \ldots, v_{n+1} \)) as it proves to be more convenient in this particular context.

It is obvious, that in order to achieve endpoint interpolation, and given the knot constraint mentioned above, we set the first (last) control point to be equal to the first (last) data point, thus giving:

\[
v_{-2} = p_1 \quad \text{and} \quad v_{n-1} = p_n
\]

In order to define the rest of the control points (from \( v_{-1} \) to \( v_{n-2} \)) we proceed as follows:

According to the definition of a B-spline curve over the knot sequence \( t_1, t_2, t_3, \ldots, t_{n-2}, t_{n-1}, t_n \) we have:

\[
v_i = f(t_{i+1}, t_{i+2}, t_{i+3}) \quad \text{where} \quad i = 0, 1, \ldots, n - 3
\]

If we now consider a DeCasteljau triangle as follows:

\[
\begin{align*}
v_{i-3} &= f(t_{i-2}, t_{i-1}, t_i) \\
v_{i-2} &= f(t_{i-1}, t_i, t_{i+1}) \\
v_{i-1} &= f(t_i, t_{i+1}, t_{i+2})
\end{align*}
\]

we can therefrom derive the following set of equations:

\[
p_i = a_{i-3}v_{i-3} + b_{i-2}v_{i-2} + c_{i-1}v_{i-1} \quad \text{for} \quad i = 3, 4, \ldots, n - 2
\]

(1)

In this set of equations, the coefficients \( a_{i-3} \), \( b_{i-2} \) and \( c_{i-1} \) depend solely on the knot values, and can therefore be considered known. The unknowns in this case are the control points \( v_j \).

In the beginning of the knot sequence, we have another DeCasteljau triangle as follows:
\[
v_{-1} = f(t_1,t_1,t_2) \\
\implies f(t_1,t_2,t_2) \\
v_0 = f(t_1,t_2,t_3) \\
\implies f(t_2,t_2,t_3) \\
v_1 = f(t_2,t_3,t_4)
\]

from which we derive (similarly as above) the following equation:

\[
p_2 = a_1 v_{-1} + b_0 v_0 + c_1 v_1
\]

(2)

At the end of the knot sequence, we have another DeCasteljau triangle as follows:

\[
v_{n-4} = f(t_{n-3},t_{n-2},t_{n-1}) \\
\implies f(t_{n-2},t_{n-1},t_{n-1}) \\
v_{n-3} = f(t_{n-2},t_{n-1},t_n) \\
\implies f(t_{n-1},t_{n-1},t_n) = p_{n-1} \\
v_{n-2} = f(t_{n-1},t_n,t_n)
\]

from which we derive (similarly as above) the following equation:

\[
p_{n-1} = a_{i-3} v_{i-3} + b_{i-2} v_{i-2} + c_{i-1} v_{i-1}
\]

(3)

If we put (1), (2) and (3) together, we form a system of equations that is underconstrained, since it has \( n - 2 \) equations and \( n \) unknowns. This means that we have two degrees of freedom, i.e. two control points to choose arbitrarily in order to formulate a well posed problem. It is most convenient for practical reasons to choose \( v_{-1}(= q_1) \) and \( v_{n-2}(= q_n) \). These are the second control point and the one before the last, and they define the shape (tangent) of the curve near the end. It is beneficial to be able to choose them, because this way we can use the degrees of freedom in order to achieve \( G^1 \) continuity between two different curves (see section 5.3 for details on how to achieve this) and later on, more
importantly, between two different surfaces.

The overall set of equations can now be written in matrix form as follows:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & \ldots & 0 \\
\vdots & a_{-1} & b_0 & c_1 & 0 & 0 & \ldots & 0 \\
0 & 0 & a_0 & b_1 & c_2 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 0 & a_{n-4} & b_{n-3} & c_{n-2} \\
0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
v_{-1} \\
v_0 \\
v_1 \\
\vdots \\
v_{n-4} \\
v_{n-3} \\
v_{n-2} \\
q_1 \\
p_2 \\
p_3 \\
\vdots \\
p_{n-1} \\
q_n
\end{bmatrix}
\]

This is a tridiagonal matrix equation. Its solution is particularly straightforward, and of linear (\(O(n)\)) complexity, which also makes it very convenient with respect to implementation. It must be noted, that \(p\), \(q\) and \(v\) here are points either in a 2D or in a 3D space, and we have to solve such a system for each dimension, so we actually have two or three such systems to solve.

### 5.2 B-spline surface interpolation

Once the problem of curve interpolation is solved, surface interpolation becomes a rather trivial task. It can easily be proven (see [21], pp 378-382) that the problem is separable. If we have an orthogonal (quadrilateral) grid of \(m \times n\) data points:

\[
d_{11} \quad d_{12} \quad \ldots \quad d_{1n} \\
d_{21} \quad d_{22} \quad \ldots \quad d_{2n} \\
\vdots \quad \vdots \quad \ldots \\
d_{m1} \quad d_{m2} \quad \ldots \quad d_{mn}
\]

we then want to produce a grid of control points such that their tensor product generates a quadrilateral surface segment. Furthermore, we want the segment to be such that each of its edges goes through one of the boundary rows or columns of the grid given above.

To achieve that, we follow a two-step procedure. First, we take each \textit{row} of data points and perform on it a curve interpolation using the procedure described in the previous section. If we do that for every row, we end up with a \(m \times (n+2)\) matrix of (control) points (because after interpolation the number of control points is the number of data points plus 2). Then, we take each \textit{column} of this matrix and do the same thing. The result is a \((m+2) \times (n+2)\) grid of control points that define a B-spline surface with the desired properties.
Figure 25: The B-spline surface interpolation problem.

Of course, the problem is entirely symmetrically separable. This means that we can start working
with each column of data points first, and then process each row of the result matrix.

5.3 Ensuring smooth continuity between successive cubic B-Splines during interpolation.

Given that we have two sets of data points, and that we want to interpolate a cubic B-spline to
each one of them, such that the curves are successive and join together smoothly (\(G^1\) continuity), the
problem can be formulated as follows:

We have a sequence of data points \(p_1, p_2, \ldots, p_n\), and another sequence of data points
\(d_1, d_2, \ldots, d_l\) such that the two sequences are successive, i.e. \(p_n = d_1 = p\).

First of all, we want the two interpolated curves to join, i.e. the end of the first to coincide with
the beginning of the second. Using the interpolation scheme described above, this means that the last
control point of the first curve, \(v_{n-1}\), must coincide with the first control point of the second curve, \(w_{-2}\),
which in turn must coincide with the first data point of the second curve. In other words, we have:

\[
 v_{n-1} = p = w_{-2}
\]
In addition to this constraint, we want them to join smoothly with $G^1$ continuity. This means that the tangent direction at the end point of the first curve must be the same as the tangent direction at the start point of the second curve.

The tangent direction at the end point of the first curve is the same as the direction of the line defined by control points $v_{n-2}$ and $v_{n-1}$, i.e., by $v_{n-2}$ and $p$.

The tangent direction at the end point of the first curve is the same as the direction of the line defined by control points $w_{-2}$ and $w_{-1}$, i.e., by $p$ and $w_{-1}$.

Point $p$ is predetermined, but (according to the interpolation scheme described above) points $v_{n-2}$ and $w_{-1}$ correspond to degrees of freedom in the interpolation problem, and have to be arbitrarily selected by the user. It becomes now obvious, that if points $v_{n-2}$ and $w_{-1}$ are chosen in such a way that the three points $p$, $v_{n-2}$ and $w_{-1}$ are colinear, then the two tangent directions where the curves join are colinear, therefore the curves are guaranteed to join smoothly.

Figure 26: A general solution to the cubic B-spline curve smooth join problem

Of course, the problem remains underconstrained. We can set those three points to be colinear on a direction $e$, but this direction is not uniquely determined. Moreover, $e$ is the direction of the tangent vectors for the two points where the curves meet (these two points coincide). The magnitudes of the tangent vectors are also not uniquely determined. They depend solely on the distances of $v_{n-2}$ and $w_{-1}$ from $p$, but these can be any distances.
To properly constrain the problem a strategy is necessary in order to empirically set these quantities (and therefore the control points $v_{n-2}$ and $w_{-1}$) such that the form of the join is semantically feasible. The control points must be set in an adaptive and automatic manner, so that they always provide a semantically feasible smooth join. Figure 27 illustrates the strategy proposed here, which seems to be working very well with body data.

![Diagram](image)

**Figure 27: A semantically feasible solution to the cubic B-spline curve smooth join problem**

The idea is to set direction and magnitudes so that the resulting curves do not over-oscillate and do not produce ripples. Somewhat surprisingly, there appears to be nothing in the literature to guide towards the solution of the problem. To achieve the desired result, we therefore look at the behaviour of the neighbouring data points that define the two curves. It appears to be optimum when direction $e$ is parallel to the direction of the line defined by data points $p_{n-1}$ and $d_2$, which are the points on either side of $p$. This guarantees that the direction of $e$ lies between the directions of tangent vectors at $p_{n-1}$ and $d_2$ in such a way that tangent direction from $p_{n-1}$ to $d_2$ changes in a ‘monotonic’ manner.

As for magnitudes, the idea is to set the distances between the joining point and the neighbouring control points according to how near/far are the neighbouring data points. The distances between the joining point and the neighbouring data points provides an estimate of the curvature of the ideal smooth curve around the joining point. In the implementation provided, these distances are calculated and then multiplied with a normalisation factor $f$ in order to give the distance values between $p$ and the
neighbouring control points. Having calculated those distances, plus the direction vector $e$, control points $v_{n-2}$ and $w_{-1}$ are now uniquely defined.

The optimum value of factor $f$ has been empirically found to be about 0.3. Higher values give very large distances and generate a 2-lobed ‘stretch’, while lower values generate concavities around the point of joining that result to a ‘pointed lobe’. These effects are illustrated on figure 28.

(a) $f=0.1$, (b) $f=0.3$, (c) $f=0.5$, (d) $f=0.8$.

Figure 28: Appearance of the smooth join for various values of the normalisation factor $f$. 
(a) $f=0.1$, (b) $f=0.3$, (c) $f=0.5$, (d) $f=0.8$. 
In case we have one curve only, that we want to be a smoothly closed loop, this technique can be used with absolutely no modification. The two curves that we join in this way do not really have to be different. We can simply regard the two first data points as the first data points of a different curve, and proceed exactly as above.

5.4 B-spline surface fitting on a smoothly closed deformed cylinder

Given the above, smoothly joining the seam on a deformed cylinder implemented as a quadrilateral B-spline surface becomes a pretty straightforward task. If we interpolate row-wise first and then column-wise in order to get the surface, all we have to do is use the ‘free’ control point selection strategy described in the previous section.

In this case, each row of data points can be regarded as a data point sequence for a smooth closed loop to be interpolated around it. All that needs to be done is to duplicate the first data point at the end of the sequence, and then proceed to applying the interpolation scheme described in section 5.1, choosing the two degrees of freedom according to the strategy described in section 5.3. After doing that for each row of data points, we can normally proceed to interpolating through each column of the result.

5.5 Smooth segment joining

The last problem that needs to be solved in order to complete the method for generating a surface around a complex object such as the human body, is the problem of making the surfaces that represent each surface join smoothly with each other. Once again, the strategy described in section 5.3 is used in the same way as in the case of smooth closing of a deformed cylinder.

For simplicity, the case presented here is the simplest one, i.e. two quadrilateral segments joining edge-to-edge, but it is not difficult to see that this can be extended to any sort of quadrilateral segments with orthogonal connectivity.

Let \( a_{ij} \) an \( m \times n \) grid of data points that define a quadrilateral surface segment A and \( b_{ij} \) an \( m \times l \) grid of points that define a second segment B.

\[
\begin{array}{cccccc}
  a_{11} & a_{12} & \ldots & a_{1n} & b_{11} & b_{12} & \ldots & b_{1l} \\
  a_{21} & a_{22} & \ldots & a_{2n} & b_{21} & b_{22} & \ldots & b_{2l} \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
  a_{m1} & a_{m2} & \ldots & a_{mn} & b_{m1} & b_{m2} & \ldots & b_{ml} \\
\end{array}
\]

We want them to join in such a way that the ‘right’ edge of A joins smoothly with the ‘left’ edge
of B. This means that $a_{jm} = b_{jm}$ for $j = 1, 2, \ldots, m$. It is obvious that, if we consider each row separately, we are back to the curve case of section 5.3. If we take the $j$-th row of A and the $j$-th row of B, we interpolate through the two sequences so as to create two curves that join smoothly. To achieve that we set the ‘free’ control points according to data points $a_{j,m-1}$ and $b_{j2}$. Once this is done for every row, we just interpolate column-wise as normal to get the surfaces.

In case we have four segments to be joined smoothly, with a connectivity like the one illustrated in figure 29, then we act absolutely similarly.

![Figure 29: Four quadrilateral segments with orthogonal connectivity](image)

First, we interpolate each segment row-wise, so as to ensure a smooth join between 1 and 2 and between 3 and 4. This breaks down to the simplest 2-segment edge-to-edge case. Then, we interpolate each segment column-wise, so as to ensure a smooth join between 1 and 3 and between 2 and 4. This also breaks down to the simplest 2-segment edge-to-edge case.

In the case that the segment has a ‘free’ edge, i.e. an edge that does not join with another edge, we still have to set the degrees of freedom. In this case, it is common practice to set the ‘free’ control points of each row/column to coincide with the endpoints of the data point sequence(s). This eliminates the degree of freedom that contributes the curvature at the boundary of the patch. In this case, curvature is uniquely defined by the rest of the data points. A typical example of ‘free’ edges are the bottom edges of the leg segment, i.e. those who define the shape of the feet at the bottom-most slice.
The pictures that follow illustrate a more complex case, the case of four segments with a branching topology. We have a ‘torso’ that is broken into two secondary segments, the ‘left torso’ and the ‘right torso’ (according to the segmentation rules described in section 3.6) as well as two ‘legs’.

Figure 30: (left) A data point grid representing a branching object. (right) Curves are fitted along rows and columns of points to show the topology. Darker curves represent the boundaries between the segments

This case includes all the simpler cases, so the way to ‘skin’ this object is the following:

- We interpolate row-wise on the left torso and the right torso (as in the simplest case). We set the degrees of freedom so that the left edge of the right torso joins smoothly with the right edge of the left torso along the ‘front seam’, and also so that the right edge of the right torso joins smoothly with the left edge of the left torso along the ‘rear seam’
- We interpolate row-wise on the left leg using the closed cylinder strategy. We do the same on the right leg.
- We interpolate column-wise on the left leg and left torso so that they join smoothly. This is again as in the simplest case.
- We interpolate column-wise on the right leg and right torso so that they join smoothly. This is again as in the simplest case. The final result can be seen in figure 32.
Figure 31: The B-spline surfaces fitted along each segment, together with the control points. Degrees of freedom have been set so as to ensure continuity between segments. It is also easy to see the 1-1 correspondence between rows and columns of points of neighbouring segments.

Figure 32: The segments put together to form the final surface.
6. Volume and surface area calculation

This chapter contains a presentation of the attempts that have been made to calculate the volume and surface area of the human body, using all forms of data that have been made available during all the other stages of processing (initial point cloud, LS-fitted horizontal B-spline curves, resampled slices and fitted surfaces). The techniques described in this chapter were implemented (unless otherwise stated) and tested on real data. However, it has not been possible to process the whole body, because of the problems associated with the top of the head and the tips of the fingers, as described in section 3.2. It was thus impossible to provide measurements for whole bodies in this report, and a claim that this objective has been achieved cannot really be made. Instead, estimates of measurements are presented here, as well as volume data acquired using the traditional methods. All these can be used for further development in the near future. Some general ideas on how to use the work presented in this chapter are included both here and in section 8.1 ('Further Work').

6.1 Volume calculation

The method proposed here is based on the slice structure of the data (horizontal B-spline curves), as well as their cylindrical topology. The idea is that, since two neighbouring slices are parallel (because they are both horizontal) and the distance between them is known, then we can approximate the volume of this part of the body that is between them by calculating the volume of the closest approximating deformed cylinder. To visualise this, we can imagine that each horizontal curve defines a ‘ribbon’ that is 5 mm wide (that is the distance between two successive rows of data). If we know the area within the curve (that is the ‘base’ of the cylinder which will from now on be called ‘enclosed area’), we can then multiply it by the width of the ribbon (that is the ‘height’ of the cylinder), and this gives us the volume of a slice. From that point on, we can add up all the slice volumes to get a measurement for the whole body volume, or we can apply on them other kinds of numerical integration as shown in section 6.1.2.

The idea for this sort of approach was based on the work of Cook et.al. [9] on volume calculation based on polyhedral approximation. The method proposed there is designed exactly for the case where data are organised in slices, but the approaches presented here are expected to be more accurate than simple polyhedral approximation, because the assumption that the object is defined by a 3rd degree surface that interpolates through the points is also taken into account.

Another approach that has been considered for the calculation of the volume was the use of the Marching Cubes algorithm, as proposed by Lorensen et. al. [8]. This is a well-established technique that
has proved to work very well. It is rather complicated, but the accuracy of the results makes up for its complexity. However, the Marching Cubes algorithm was finally not used, as it is rather inappropriate given the structure of the data here. More specifically, this algorithm is design to work with voxel data, not with points. The processing of the dataset in order to convert point data to voxel data is not a straightforward task, and moving towards that direction would be rather costly in terms of time and design/programming effort. An alternative approach that has also been considered was to use the two-dimensional version of the Marching Cubes (‘Marching Squares’) to calculate the enclosed areas of the slices, but its implementation would require an amount of programming effort that was beyond the scope of this project.

Part of the work for body volume calculation involved assembling a number of subjects and measuring their volumes using traditional methods (immersion and pressure pod [2]). It was originally intended for these data to be used so that a comparison could be made against the results of the techniques proposed here. Six subjects were used in total. The measurements took place at the Dunn Nutrition Research Center of the Addenbrooke Hospital in Cambridge, where all the facilities and equipment for such a task can be found.

Each subject’s volume was measured using each method twice, thus giving a total of four measurements per subject. The results of these measurements are given in the following table.

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>Mid-tidal lung volume</th>
<th>Immersion volume</th>
<th>Bod-Pod volume (excl. lungs)</th>
<th>Total volume (immersion)</th>
<th>Total volume (Bod-Pod)</th>
<th>Difference between measurements</th>
<th>Mean between volumes</th>
<th>%ERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (LD)</td>
<td>3.900</td>
<td>51.290</td>
<td>51.090</td>
<td>55.190</td>
<td>54.990</td>
<td>0.200</td>
<td>55.090</td>
<td>0.363</td>
</tr>
<tr>
<td></td>
<td>3.500</td>
<td>51.140</td>
<td>50.800</td>
<td>54.640</td>
<td>54.300</td>
<td>0.340</td>
<td>54.470</td>
<td>0.624</td>
</tr>
<tr>
<td>2 (YD)</td>
<td>3.700</td>
<td>64.400</td>
<td>64.590</td>
<td>68.100</td>
<td>68.290</td>
<td>0.190</td>
<td>68.195</td>
<td>0.279</td>
</tr>
<tr>
<td></td>
<td>3.600</td>
<td>64.900</td>
<td>64.530</td>
<td>68.500</td>
<td>68.130</td>
<td>0.370</td>
<td>68.315</td>
<td>0.542</td>
</tr>
<tr>
<td>3 (DY)</td>
<td>3.100</td>
<td>55.230</td>
<td>53.850</td>
<td>58.330</td>
<td>56.950</td>
<td>1.380</td>
<td>57.640</td>
<td>2.394</td>
</tr>
<tr>
<td></td>
<td>4.500</td>
<td>55.070</td>
<td>54.550</td>
<td>59.570</td>
<td>59.050</td>
<td>0.520</td>
<td>59.310</td>
<td>0.877</td>
</tr>
<tr>
<td>4 (SB)</td>
<td>3.500</td>
<td>49.560</td>
<td>48.940</td>
<td>53.060</td>
<td>52.440</td>
<td>0.620</td>
<td>52.750</td>
<td>1.175</td>
</tr>
<tr>
<td></td>
<td>3.100</td>
<td>49.180</td>
<td>48.890</td>
<td>52.280</td>
<td>51.990</td>
<td>0.290</td>
<td>52.135</td>
<td>0.556</td>
</tr>
<tr>
<td>5 (ED)</td>
<td>4.200</td>
<td>59.580</td>
<td>59.180</td>
<td>63.780</td>
<td>63.380</td>
<td>0.400</td>
<td>63.580</td>
<td>0.629</td>
</tr>
<tr>
<td></td>
<td>4.100</td>
<td>59.410</td>
<td>59.120</td>
<td>63.510</td>
<td>63.220</td>
<td>0.290</td>
<td>63.365</td>
<td>0.458</td>
</tr>
<tr>
<td>6 (DC)</td>
<td>4.600</td>
<td>60.890</td>
<td>60.830</td>
<td>65.490</td>
<td>65.430</td>
<td>0.060</td>
<td>65.460</td>
<td>0.092</td>
</tr>
<tr>
<td></td>
<td>4.800</td>
<td>61.120</td>
<td>61.010</td>
<td>65.920</td>
<td>65.810</td>
<td>0.110</td>
<td>65.865</td>
<td>0.167</td>
</tr>
</tbody>
</table>

All subjects were also scanned the day before the visit to Addenbrooke Hospital, wearing the same garments (underwear or swimsuits). They were also instructed to follow a moderate diet during the 24 hour period between the scanning and the conventional measurements. These rules had to be followed so that their volume would remain almost unchanged. Otherwise, it would not be feasible to use the
results for comparing the various methods with any great confidence as to the accuracy of the comparison.

Since automatic volume calculation has not been implemented for the whole body, these data have not really been used. However, they can always be made available for future use, as scans of these subjects have been retained on file.

6.1.1 Calculation of the enclosed area in a closed B-spline curve.

In this section, two methods for the calculation of the enclosed area of a slice are presented. Both of them have been implemented and tested, and their behaviour is analysed here. First, a description of the methods is given, together with the underlying mathematics, and a comparison between them follows immediately afterwards.

The problem we have is that we want to calculate the area that is enclosed within a closed curve whose points are all co-planar. This curve is represented as a 3rd degree B-spline. The curve might have concavities, but it is not self intersecting.

Both methods proposed here are based on the same logic. The area inside the curve is subdivided into smaller entities of known shape. Since the area of each of these entities is easy to calculate, we can therefore calculate it and sum the results. We can then iteratively proceed to a further subdivision and repeat the calculation, until the value of the sum begins to converge in a numerical sense\(^1\).

The problem now reduces to finding out an optimal way of subdividing the area inside the curve. In this context, ‘optimum’ is the subdivision strategy that:

- gives the fastest, preferably monotonic convergence
- converges to a value that is as close to the actual area inside the curve as possible.

The two subdivision strategies proposed are: triangulation and Simpson’s rule. Both of them require resampling of the curve, with the resampling rate increasing at each iteration.

**Triangulation**

The triangulation method works as follows:

First we take a number of samples (say \(n\)) on the curve. It is best to take the samples by evaluating the B-spline at uniform intervals of the parameter range, so that we get more dense samples where curvature (and therefore detail level) is higher. This creates a point sequence of the form:

\[ s_{0}, s_{1}, s_{2}, \ldots, s_{n} \]

\(^1\) Either the difference between two successive values becomes smaller than a predetermined accuracy threshold \(e\), or the number of iterations exceeds a user defined maximum.
Then, we choose a point $s$ on the plane where the curve is. This can be any point, as it will be proven later. However, in practice we choose $s$ to be either one of the samples or, even better, we choose it to be the centroid of the curve.

![Figure 33: Subdivision of the area inside a closed curve by triangulation. Point $s$ can be either one of the samples (left) or the centroid of the curve (right)](image)

This collection of points defines a sequence of triangles of the form $(s \ s_{i-1} \ s_i)$, with $i = 1, \ldots, n$.

Calculation of the area for each one of those triangles is a trivial operation, which only requires the calculation of a determinant. If we choose $s$ to be the centroid of the curve, the angle of the triangle at $s$ tends to be less sharp, which makes calculation of the determinant more stable. This is why we prefer to choose the centroid of the curve instead of one of the samples.

If we sum the areas of triangles, we get the area inside the curve minus the sum $e$ of the small residues $e_i$ defined by:

- The line segment between points $s_{i-1}$ and $s_i$
- The segment of the curve between points $s_{i-1}$ and $s_i$

If we now increase the number of samples, we get more and smaller triangles. Then, the area of each $e_i$ is smaller in such a way that their sum $e$ is smaller, so the sum of the areas of triangles approaches the actual area inside the curve.

This method is not affected by concavities in the curve. If (say) the curve runs clockwise, then the value of the determinants will be positive except where concavities occur (figure 34), in such a way that overlapping triangles are subtracted from each other and finally a feasible approximation of the actual area is returned.
Figure 34: Sample tolerance of triangulation against concavities. In the picture, shaded triangles have a negative value for their area, non-shaded triangles have a positive value. The algebraic sum of areas is such that common regions of overlapping triangles are cancel each other.

**Simpson’s rule**

This method has been tried as an alternative to triangulation, hoping that it would lead to faster convergence and more accurate results. This expectation was based on the method’s property of being mathematically exact for samples that represent curves of up to 3rd degree (just like the curves that have been used throughout the whole project).

First, a description of Simpson’s rule in its general form is presented, and then a demonstration of how the mathematical model applies to the case a closed B-spline.

In the general case, we seek to calculate the integral (i.e. the area enclosed between a curve and the x-axis):

$$E = \int_{x_0}^{x_1} f(x)dx$$

Where $f$ is a function that can be approximated by a 3rd degree polynomial. What we are given is three values $f_i$ of the function for the corresponding values $x_i$ of the input parameter (where $i = 0,1,2$).
The values $x_i$ do not need to be monotonically sorted. However they must all be different:

![Graph showing values $x_i$ and corresponding $f(x)$]

Given that $f(x)$ is a 3rd degree function, we approximate it using the 2nd degree Lagrange polynomial $p(x)$:

$$p(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f_2$$

(This is why the values $x_i$ need to be all different. If they are not, a division by zero will occur and further calculations are impossible. Moreover, it can easily be proven that, if they are in increasing order, then the value of $E$ will be positive. Otherwise, if they are in decreasing order, then the value of $E$ will be negative. This is a useful feature, as it guarantees that the method is not affected by curve concavities, as will be described later in this chapter.)

Therefore, we have:

$$E = \int_{x_0}^{x_2} f(x)dx = \int_{x_0}^{x_2} p(x)dx$$

Now, if we use the notation:

$$h_1 = x_1 - x_0$$
$$h_2 = x_2 - x_1$$
$$m = x_2 - x_0 = h_1 + h_2$$
we can rewrite $p(x)$ in the standard Bernstein basis form, as:

$$p(x) = b_0 x^2 + b_1 x + b_2$$

with:

$$b_0 = a_0 + a_1 + a_2$$
$$b_1 = -\left[a_0(x_1 + x_2) + a_1(x_0 + x_2) + a_2(x_0 + x_1)\right]$$
$$b_2 = a_0 x_1 x_2 + a_1 x_0 x_2 + a_2 x_0 x_1$$

where: $a_0 = \frac{f_0}{m h_1}$, $a_1 = \frac{-f_1}{h_1 h_2}$, $a_2 = \frac{f_2}{m h_2}$

Therefore, the integral can now be calculated as:

$$E = \int_{x_0}^{x_2} p(x) dx = b_0 \int_{x_0}^{x_2} x^2 dx + b_1 \int_{x_0}^{x_2} x dx + b_2 \int_{x_0}^{x_2} dx$$

The three new integrals are trivial cases of polynomial integration, calculated using standard formulae, so we have:

$$E = b_0 \left[\frac{x_2^3 - x_0^3}{3}\right] + b_1 \left[\frac{x_2^2 - x_0^2}{2}\right] + b_2 (x_2 - x_0)$$

In the special case where $x_0 \leq x_1 \leq x_2$ and $x_1 - x_0 = x_2 - x_1 = h$, by following the same steps we get:

$$E = \frac{h}{3} (f_0 + 4 f_1 + f_2)$$

which is the commonly used version of the Simpson’s rule.

To use Simpson’s rule for a closed B-spline, we proceed as follows:

First we take a number of samples (say $n$) on the curve, creating a point sequence of the form:

$$s_0, s_1, s_2, ..., s_n.$$
Again, it is best to take the samples by evaluating the B-spline at uniform intervals of the parameter range, so that we get more dense samples where curvature is higher.

However, the sequence must now be such that the number of successive intervals between the samples is even so that we can apply Simpson’s rule to pairs of intervals. The number of samples must therefore be odd which, in turn, means that \( n \) must be even since we start numbering from zero.

After taking the samples, we consider a line on the plane. This can be any line, but in order to simplify the equations it is best to consider the x-axis (see also figure 35). Now, for \( i = 0, 1, 2, \ldots, (n/2) - 1 \) we set:

\[
\begin{align*}
  x_0 &= s_{2i}, & f_0 &= s_{2i}, \\
  x_1 &= s_{2i+1}, & f_1 &= s_{2i+1}, \\
  x_2 &= s_{2i+2}, & f_2 &= s_{2i+2},
\end{align*}
\]

and then we can apply Simpson’s rule on the \( x_0, x_1, x_2, f_0, f_1, f_2 \) to calculate the area \( S_i \) of the ‘stripe’ that is defined by:

- The line segment between points \((x_0, 0)\) and \((x_2, 0)\)
- The curve segment between points \(s_{2i}\) and \(s_{2i+2}\)

Figure 35: Calculation of the enclosed area by subdividing into stripes and applying Simpson’s rule on them
After we have calculated the \( e_i \) for every \( i \), we sum them up. The sum:

\[
S = \sum_{i=0}^{(n/2)-1} S_i
\]

is an estimate of the actual area \( S \) enclosed inside the curve. This is because the value and sign of each \( S_i \) depends on the sorting order of \( x_0, x_1 \) and \( x_2 \), so that whenever we have overlapping stripes only the area within the curve contributes to the sum (just once) and everything else is cancelled. This also ensures that the method will work even if the curve has concavities.

If we now increase the number of samples (by 2, so that it still remains odd) and repeat the process, we expect to get a more accurate measurement since for each stripe the difference between the segment of the curve that delimits the stripe and the segment of the Lagrange polynomial that we interpolate through the samples tends to be smaller. Therefore, we can expect that if we keep iterating the process, each time increasing the number of samples, the value of the sum \( S \) will converge to the value of the area inside the curve.

A comparison between the two techniques

Both techniques seem to be quite appropriate for the particular application domain. The results they return are accurate for up to two decimal places, corresponding to an accuracy of the order of 0.01mm\(^2\), and they both converge to that accuracy within a few hundreds of iterations (see figure 36). They have been tested with a number of closed B-splines (about 30), some of which were taken from actual scans while others were designed by hand (to investigate the behaviour for multiple concavities and various levels of curvature), and they demonstrated a consistent behaviour regardless of the type of curve\(^2\).

In 90% of the cases, Simpson’s method gave a slightly higher final value than the triangulation method (for the same curve). However, this difference has always been observed after the second decimal place.

Figure 36 shows the convergence behaviour of the two methods for the sample curve shown on figures 33 and 35. As one can see, there is convergence in both cases. Graphs in figure 36 show the value of the sum \( S \) (in mm\(^2\)) for the first 200 iterations. For the example shown, desired convergence accuracy was set to 4 decimal digits for both techniques. In the first case (triangulation), the algorithm converged after 721 iterations to a value of \( S = 4858.492187\)mm\(^2\). In the second case (Simpson’s rule), the algorithm converged after 748 iterations to a value of \( S = 4858.497883\)mm\(^2\). Both methods are fast

\(^2\) Provided, of course, that the curves did not have any properties that conflicted with the assumptions mentioned above.
enough for the purposes of a body modelling application. On a Silicon Graphics workstation, the process of calculating the enclosed area of a curve takes a few milliseconds, it is therefore expected that in case of a whole body calculation will take approximately 4 to 5 seconds.

\[ Figure \; 36: \; Convergence \; behaviour \; of \; the \; iterative \; methods \; for \; the \; calculation \; of \; the \; enclosed \; area. \]

(a) Triangulation. (b) Simpson’s rule.

The unexpected difference in the convergence behaviour of the two techniques is that
triangulation demonstrates a much more stable behaviour. In fact it seems to be better for the task, despite requiring the calculation of determinants for triangles with very sharp angles, because it always behaves in a predictable way. The first iteration gives a value that is smaller than the actual area, and from then on that value increases monotonically with every iteration, as the residue becomes smaller (figure 36a).

On the other hand, Simpson’s rule does not behave as well as triangulation. It takes approximately 30 iterations for the algorithm to start demonstrating some stability, but even beyond that point, in some iterations the value diverges significantly from the convergence level (this is represented by the ‘spikes’ in the chart, figure 36b). This is rather dangerous, as it can affect convergence (introduce the necessity for more iteration), and it can also lead to inaccurate results (e.g. in the case where the user specifies a maximum of 600 iteration and there is a ‘spike’ - inconsistent value - on the 600th).

For these reasons, it is suggested that, although Simpson’s rule would appear theoretically to be better for the task, triangulation should be preferred owing to its more stable characteristics.

6.1.2 Volume calculation by integrating slice areas

After calculating the enclosed area for each slice of the body, the results must be used to provide a measurement for the volume. This is a problem of numerical integration, and there are multiple approaches towards its solution. In all cases, each segment must be processed separately (as a deformed cylinder), and then the volumes of each segment can be summed to provide a volume measurement for the whole body.

The first approach is to follow a strategy which is analogous to the trapezoid rule for integrating a function. If a segment has \( n \) slices with enclosed areas \( a_1, a_2, \ldots, a_n \) and \( d_i \) is the distance between the \( i \)-th and the \((i+1)\)-th slice, then we can estimate the volume \( v_i \) between them as:

\[
v_i = \frac{a_i + a_{i+1}}{2} d_i
\]

The volume of the segment can then be acquired by summing, i.e: \( V = \sum_{i=1}^{n-1} v_i \)

The second approach is based on the assumption that the wall of the deformed cylinder is a 3rd degree surface that interpolates through the slice curves, therefore the strategy is analogous to Simpson’s rule. If \( n \) is an odd number, then we can take three successive slices at a time, i.e the \( i \)-th the \((i+1)\)-th and the \((i+2)\)-th slice. If the height \((z\)-coordinate\) of each slice is \( z_i, z_{i+1} \) and \( z_{i+2} \) respectively, we can then
calculate the volume \( v_i \) between the \( i \)-th and the \( (i+2) \)-th slice by using the general form of Simpson’s formula presented previously in this chapter, by setting:

\[
\begin{align*}
x_0 &= z_i & f_0 &= a_i \\
x_1 &= z_{i+1} & f_1 &= a_{i+1} \\
x_2 &= z_{i+2} & f_2 &= a_{i+2}
\end{align*}
\]

Then, the volume of the segment can be acquired by summing, i.e:

\[
V = \sum v_i \quad \text{but now with } i = 1, 3, 5, ..., n-2
\]

If the segment has an even number of slices, we can proceed as above for the first \( n-1 \) slices, and add to the result the estimate of the volume between the \( (n-1) \)-th and the \( n \)-th slice. This estimate can be acquired using the trapezoid rule.

Since the solid whose volume we seek to calculate is defined by a 3rd degree surface that interpolates through the slice curves, the second approach (using Simpson’s rule) is expected to be more accurate, because it takes the form of the surface into account, whereas the first approach (trapezoid) does not.

### 6.2 Surface area calculation

As well as with the methods proposed for volume calculations, the method proposed here is also based on the slice structure of the data (horizontal B-spline curves), as well as their cylindrical topology. Again, the idea is that, since two neighbouring slices are parallel (because they are both horizontal) and the distance between them is known, we can approximate the volume of this part of the body that is between them by calculating the surface of the wall of the closest approximating deformed cylinder. To visualise this, we can again imagine that each horizontal curve defines a ‘ribbon’ that is 5 mm wide (that is the distance between two successive rows of data). If we know the length of the curve, we can then multiply it by the width of the ribbon (that is the ‘height’ of the cylinder), and this gives us an approximation of the ‘surface area of a slice’\(^3\). From that point on, we can add up all the slice surface areas to get a measurement for the whole body surface area. Alternatively or we can take a weighted sum of the slice lengths (effectively a numerical integration) as shown in 6.2.2.
6.2.1 Slice length calculation

The method for numerically calculating the length of a curve is straightforward and long-established ([19] - p461). Again, we have to take a sequence of samples (say \(n\)) on the curve. Once again, it is best to take the samples \(s_0, s_1, s_2, \ldots, s_n\) by evaluating the B-spline at uniform intervals of the parameter range, so that we get more dense samples where curvature (and therefore detail level) is higher.

For every two successive samples in the sequence, we calculate the Euclidean distance:

\[
d_i = \sqrt{(s_{i+1} - s_i)^2 + (s_{i+1} - s_i)^2} \quad \text{with} \quad i = 0, 1, \ldots, n-1
\]

We can then get an estimate of the curve length \(L\) by adding up the \(d_i\):

\[
L = \sum_{i=0}^{n-1} d_i
\]

The error between \(L\) and the actual length of the curve is the sum of differences between:

- The length of the line segment between points \(s_i\) and \(s_{i+1}\)
- The length of the segment of the curve between points \(s_i\) and \(s_{i+1}\)

(for every \(i = 0, 1, \ldots, n-1\))

If we now increase the number of samples, then this error will become smaller. In fact, if we iterate the process, each time increasing the number of samples, then the line segments between successive samples will tend to be infinitely small, and the error will tend to zero and the algorithm will converge to the actual length of the curve. Once again, we can thus iterate the process until the length value \(L\) converges in a numerical sense. The convergence behaviour of this method is very stable, and can easily provide an accuracy of up to 6 decimal places for the typical curves of interest here, usually within 70 iterations or so.

6.2.2 Surface area calculation by integrating slice lengths

After calculating the length for each slice of the body, the results must be used to provide a measurement for the body surface area. This is a problem of numerical integration, just like the problem of calculating the volume given the enclosed area for each slice, and there are several approaches towards its solution. In all cases, each segment must be processed separately (as a deformed cylinder), and then the surface area of each segment can be summed to provide a surface area measurement for the whole body.

---

3 However, this approximation is rather poor for successive slices that are significantly different to each other. This case is discussed in more detail later in this chapter.
The first approach is to follow a strategy which is analogous to the rectangle rule for integrating a function. If a segment has \( n \) slices with lengths \( l_1, l_2, \ldots, l_n \) and \( d_i \) is the distance between the \( i \)-th and the \( (i+1) \)-th slice, then we can estimate the surface area \( A_i \) of each slice as the surface area of a ‘ribbon’:

\[
A_i = l_id_i \quad \text{for} \ i = 1, 2, \ldots, n
\]

i.e. the surface area of the wall of a deformed cylinder that is locally defined around the slice.

The surface area of the segment can then be acquired by summing, i.e.:

\[
A = \sum_{i=1}^{n} A_i
\]

Some extra care must be taken to avoid a certain artefact in the measurement. The top slice of a segment is the same as the bottom slice of the segment right above it, and vice versa. If we just add the surface areas of all segments, then the areas \( v_i \) that correspond to boundary slices will be included twice in the overall sum. Therefore, they must either be subtracted from the final (whole body) sum, or not be included in the partial (segment) sums but added afterwards, to ensure that they contribute to the total area only once.

There is a problem with the accuracy of this method. Most of the successive slice curves (e.g. torso, legs) are fairly similar to each other. In this case, the part of the surface between them is well approximated by a ribbon. However, for other areas of the body, such as the shoulders, the chin, or the top of the head, successive slices can be significantly different, because the surface of the body (given the particular scanning posture described in chapter 3) tends to be more horizontal than vertical. This means that the surface between two successive slice curves is very poorly approximated by a ribbon. It has a ribbon topology, but the geometry is much closer to a cone frustum than to a ribbon. The aforementioned method does not take this into account. However, there can be a slight modification so that this type of geometry is taken into account.

The suggestion here is to approximate the surface between the two curves with the frustum of a cone. In this case, we need both the length \( l_i \) and the enclosed area \( a_i \) of each curve. In this case the area \( A_i \) between the \( i \)-th and the \( (i+1) \)-th curve can be approximated by the formula that gives the area of a cone frustum (excluding the top and bottom circles):

\[
A_i = \sqrt{(a_i + a_{i+1})^2 + d_i^2\left(\frac{l_i + l_{i+1}}{2}\right)^2} \quad \text{for} \ i = 1, 2, \ldots, n-1
\]

The surface area of the segment can then be acquired by summing, i.e.:

\[
A = \sum_{i=1}^{n-1} A_i
\]

As opposed to the case of ribbon approximation, this method introduces no artefacts due to overlapping when the areas of each segment are summed.
The second approach is based on the assumption that the wall of the deformed cylinder is a 3rd degree surface that interpolates through the slice curves, again using a strategy analogous to Simpson’s rule. If \( n \) is an odd number, then we can take three successive slices at a time, i.e the \( i \)-th the \((i+1)\)-th and the \((i+2)\)-th slice. If the height (z-coordinate) of each slice is \( z_i \), \( z_{i+1} \) and \( z_{i+2} \) respectively, we can then calculate the surface area \( A_i \) between the \( i \)-th and the \((i+2)\)-th slice by using the general form of Simpson’s formula presented previously in this chapter, by setting (as in the case of body volume):

\[
\begin{align*}
x_0 &= z_i, & f_0 = l_i, \\
x_1 &= z_{i+1}, & f_1 = l_{i+1}, \\
x_2 &= z_{i+2}, & f_2 = l_{i+2}
\end{align*}
\]

Then, the surface area of the segment can be acquired by summing, i.e:

\[
A = \sum A_i \quad \text{with} \quad i = 1, 3, 5, ..., n-2
\]

If the segment has an even number of slices, we can proceed as above for the first \( n-1 \) slices, and add to the result the estimate of the surface area of the ‘ribbon’ that corresponds to the \( n \)-th slice. This estimate can be acquired using the trapezoid rule as in the previous approach.

It must be noted that there is nothing special happening where the segments branch. This holds true for any of the aforementioned methods. That is because with the type of segmentation used, all slices have the same topology, even when there are branches. One might consider, for example, the bottom slice of the torso. The topology of this slice is illustrated in figure 37.

Figure 37: Topology of the bottom-most slice of the torso as defined by the two bottom-most curves. The area between the curves is deformed, but still a ‘ribbon’. All the aforementioned calculation techniques can therefore be used without any special treatment.

6.2.3 Alternative proposal: Surface area calculation by recursive subdivision of B-spline surfaces
A completely different approach to the problem of calculating the body surface area, is to use an adaptive algorithm. This can be done by recursively calculating the surface area for each (secondary) segment, then adding the results for each secondary segment to get a measurement of the surface area for the whole body. Calculation of the surface area for each B-spline surface (for each secondary segment) can be done as follows:

The B-spline surface is parametrically defined in a 2D parametric space. Let $u$ and $v$ be the two parameters, and also let $u$ range from $u_1$ to $u_2$ and $v$ range from $v_1$ to $v_2$. If $B(u_0,v_0)$ is the point on the surface that we get by evaluating the surface’s parametric expression for the values $u_0$ and $v_0$ of the parameters, then we can consider the following grid of points:

$$
B(u_1,v_1) \quad B\left(\frac{u_1 + u_2}{2}, v_1\right) \quad B(u_2,v_1) \\
B(u_1,\frac{v_1 + v_2}{2}) \quad B\left(\frac{u_1 + u_2}{2}, \frac{v_1 + v_2}{2}\right) \quad B(u_2,\frac{v_1 + v_2}{2}) \\
B(u_1,v_2) \quad B\left(\frac{u_1 + u_2}{2}, v_2\right) \quad B(u_2,v_2)
$$

If all these points are coplanar, and the points in each one of the top and bottom rows and the leftmost and rightmost columns are collinear, then we can consider the surface to be a degenerate case, a coplanar quadrilateral whose area is trivial to calculate.

Otherwise, we can subdivide the surface to four smaller surfaces, based on the points above (i.e. subdivide by halving the parameter space each time. Then each one of these four surfaces can be processed as above (check for co-linearity and co-planarity, if the criterion is satisfied calculate the areas of the quadrilaterals otherwise subdivide again), and the four results can be added up to give the total area. Thus, we can recursively subdivide the area until we reach a level where all subdivisions are so small that they can be regarded as planar faces. The advantage of this algorithm is that it is adaptive, i.e. there are more calculations for parts of the surface with higher detail and less calculations otherwise. It is easy to devise numerical (distance) criteria to determine whether points are collinear or coplanar [26], and by setting distance thresholds we can adjust the accuracy of the algorithm to the desired level. Accuracy of this method is limited only by the accuracy (word length) of the hardware.

There is also another, simpler way to subdivide the surface. This is based on the convex hull property of B-spline segments [35]. Every B-spline segment (curve or surface) is enclosed into the convex hull defined by its control points. Therefore, if all four control points of a curve segment are collinear, the segment is a straight line segment defined by the first and last control point. This extends to the
surface case as follows: A B-spline surface patch is defined by a 4x4 grid of control points. If all 16 control points of a B-spline surface patch are co-planar, and the control points in each of the outermost rows and columns of the grid are co-linear, then the patch is a quadrilateral and the calculation of its area is a trivial task.

Based on this property, we can recursively subdivide the surface, in each step checking the co-linearity and the co-planarity of the control points themselves, without evaluating the surface.

However, this algorithm is rather complicated in both versions. In both cases, implementation requires the use of complex data structures such as quad-trees in order to achieve an acceptable performance and make good use of computational resources (by avoiding explicitly to call a function recursively), and this is why a decision was made not to implement it, as it would be rather time consuming given the short-term nature of this project. An alternative implementation idea (courtesy of Joao Oliveira) was to take advantage of the code already used by OpenGL for rendering B-spline surfaces. OpenGL uses exactly the same algorithm for rendering a B-spline surface, i.e. it subdivides it until each patch can be small enough to be rendered as a polygon face. If there was a way to retrieve the values of the vertices as OpenGL runs the rendering code, it would be quite easy to calculate the areas of the facets and then sum them to get the final surface area. However, such a facility does not seem to be available for the user of the library, as the values of the vertices are immediately passed on to the rendering pipeline without being made accessible by the user.

The problem is of purely technical nature, and it could be a challenging task for further development.
7. Results

The results for the individual components have already been discussed. In this chapter, the output from the complete work is considered. First, a series of illustrations that depict the steps in the complete steps are presented. Then, an attempt is made to assess the quality of the results, followed by a discussion on the strengths and limitations of the proposed algorithm.

7.1 Output of the algorithm

The body surface, after processing the point set by all of the aforementioned procedures is shown in figures 38 and 39. Figure 38 illustrates the stages of processing. For the particular illustration, the sampling scheme used in figure 38 was the one using uniform parametric intervals with re-ordering and calibration (section 4.1). Figure 39 illustrates a closer view of the torso area, and this time the sampling schemes used were: (i) for the torso: intersection with cord bisector, followed by mid-bust refining, and (ii) for the legs: bounding box. In figure 39, the shoulder area has been included to the torso, with the arms ignored so that both shoulders and torso were incorporated into a single primary segment with the topology of a deformed cylinder. This was done in order to investigate the behaviour of the algorithm (using the particular sampling schemes) in the bust area, since this is of interest for some clothing design applications. It was also done in order to show the tolerance of the algorithm against poorly fit curves (see also section 4.2). Figure 40 shows the set of fitted curves from which the surface shown on figure 39 was generated. These curves are very poorly fitted around the armpits. However, the final surface can still be said to be true to the original data.

The quality of results for the leg in the case of uniform parametric resampling has already been discussed in chapter 4. Figure 41 shows the resulting surface for the same leg using some of the other sampling schemes, namely for intersection with cord bisector and for bounding box. There are still problems in the final surface for both of these methods. These problems and their sources are discussed in detail in the next section.

As for the numerical results for volume and surface area calculations, they are not presented in this report. The reason for this is that, since the skinning of the body was not finally completed, all measurements would correspond to the part of the body that has been skinned. Furthermore, since the body is segmented according to its geometry, and not according to its anatomical semantics, it is not possible to use the heuristics that are available in the literature (e.g. [33],[34]) in order to estimate the whole volume and surface area from the partial results. Therefore, presenting any numerical results would be meaningless, since they cannot be assessed in terms of their accuracy and reliability. Evaluation of the proposed methods is an issue to be addressed.
Figure 38: The stages of the complete algorithm from the original point set to the final 'skin'. From left to right: original point set, point set with LS-fitted B-spline curves along slices, sampled slices, the grid of samples (now having the desired connectivity), draft surface (created by triangulation of the sample grid), final B-spline surface.
7.2 Analysis and critique of the results

The final surface is, to a great extent, true to the original data. It can also be clearly seen that it is generally unaffected by outliers, however meaningful details such as the ‘pit’ around the navel and the concavities around the knees, are still preserved. The concavity along the spine is also a feature that is preserved in the final surface. It is only interrupted at the armpit level, because the subject was scanned in her underwear, so this is also a preserved detail. All segments join seamlessly and the surface is smooth everywhere. The only features that have been completely lost are the natural cusps of the body (more frequent on obese subjects), but this behaviour was predictable since this method emphasises on surface
smoothness and is inherently incapable of modelling discontinuities. To achieve the latter, the whole algorithm would need to be redesigned so that discontinuities can be detected at the very early stages of processing and related information can be retained throughout the process, in order to contribute to the final surface interpolation stage.

Figure 40: The set of LS-fitted B-splines from which the surface of figure 39 was created. Points indicate the original data set as captured by the scanner.

The effect of outliers, although being rather strong for some of the slice curves (figure 40), is significantly reduced in the final surface. However, outliers still do have some influence on the surface.

In general, surface quality tends to be better for the torso than for the legs. This is because none of the proposed resampling strategies seems to be good enough for the geometry of the legs. It can be
seen in figure 41 that each strategy manages to solve some problems, but generates others. The scheme that involves intersection with cord bisector (section 4.4) gives a reliable surface for the part of the leg that extends from groin to ankle, keeping details such as the shape of the kneecap. However, due to the problem mentioned in section 4.6 (poor and uneven resampling as the slices become bigger), the whole front part of the foot is missed out. This means it is generally a good method if feet are not of interest, but indicates that maybe feet should be treated as separate segments, because their topology is the same as that of the legs, but their geometry is radically different.

![Leg reconstruction using intersection with cord bisector (left) and bounding box (right) for resampling the slices.](image)

Figure 41: Leg reconstruction using intersection with cord bisector (left) and bounding box (right) for resampling the slices.

The bounding box approach was designed in an attempt to overcome this problem, however it has been rather unsuccessful. Not only is the shape of the feet still lost due to the very poor fitting of the curves in this area (self intersecting and sometimes not closed), but the surface for the leg is also ruined because helical distributions are introduced.
8. Conclusions

In this chapter, the overall quality of the work presented in this report is assessed, and directions towards further development are presented.

8.1 Further work

The further work detailed here is concerned mainly with areas of work that were not touched but are ultimately required for the completion of our body models. However, in addition, some specific problems and enhancements related to the individual algorithms but not discussed in details in previous chapters are also considered.

The quality of fit for least-squares B-spline curves

The least-squares fitting of curves along slices of data as developed by Elliot West is a very crucial component of the whole method, because it handles a number of problems that would otherwise require vast amounts of design and programming effort to be overcome. However, the method in its current implementation suffers from a number of problems that affect the final surface. More specifically, the curves produced are not always closed. The two ends are always close to each other (a few millimetres in the worst case), but they still remain open. Moreover, even if they are closed, there is no constraint to force them to join smoothly in all cases. Sampling can generally help get around this problem, however the final surface would be much more reliable if the curves followed this smoothness constraint around their ends. In addition to that, although the method is highly tolerant against outliers for most of the cases, there are occasions where a single outlier can introduce extremely large artefacts. This suggests that either the cleaning techniques should be reconsidered, or that the values of some of the heuristically adjusted parameters of the method need to be reconsidered and refined.

The use of more effective resampling strategies

The problems of the existing resampling techniques have already been discussed in detail. The idea of resampling proves to be very effective as it greatly simplifies the points-to-surface problem when the initial point set is unorganised (unstructured). However, an optimal way to perform sampling has not been achieved yet, unless manual stages are included in the procedure (e.g. the LASS system [5]). However, one of the requirements is that everything should be carried out completely automatically without user intervention at any stage of processing. The ideal solution would be to sample each slice at equal angles around a principal axis. This is the approach of the LASS method, which works very well if only the torso
is of interest. However, when the whole body is considered, the semantics of this approach do not scale easily for the case of branching. However, the approach suggests a solution that involves setting sample points where all slices intersect a single plane (this guarantees that no helical geometries will occur), and it would be interesting to investigate how this plane should be defined for the case of a complex, multi-segment object such as the human body.

Integration of arms and head to the proposed scheme

The arms and head are currently excluded from the whole procedure. However, arm and head data (excluding fingertips for the first and top for the latter) can already be processed in exactly the same way as the legs and torso, and surfaces could be generated in a similar manner. These surfaces can be easily connected to the existing skinned segments, exactly in the same way as legs are connected to the torso.

The singularities at the top of head and the fingertips

Once all of the above have been achieved, the only problem that remains to be solved is the closing of the surface at the top of the head and at the ends of the arms. This cannot be done using quadrilateral B-spline patches, for the reasons that have been explained in chapter 5. Therefore, some other surface model needs to be considered. One of the approaches that seems to be the most appropriate is to model these parts of the body as deformed hemispheres. This can be done by locating the top point of the head and constructing curves that start from that top point and extend radially to the sample points of the top slice (in the same way that meridians on a globe join at the poles). These curves, together with the respective segments of the top slice curve can define triangular B-spline surface patches, and smooth joining of those patches can be guaranteed since triangular patches also incorporate degrees of freedom that can be manipulated in order to achieve desired tangency at their boundaries.

Volume and surface area calculations for the whole body

After the task of producing a complete surface for the whole body has been accomplished, the proposed volume and surface area calculation methods can be combined with the full body model. This will enable the production of volume and surface area estimates for a whole human body, from which it will be possible reliably to test the accuracy of the proposed methods. So far comparison of the proposed methods with other physical measurements, for example of the body volume has not been possible. A complete, rigorous evaluation of the techniques developed has thus not been carried out, although this
work will be carried further into a new project where this will be done. Such a project is already scheduled to begin in October 1998 in the Computer Science Department, funded by the Wellcome Trust.

8.2 Final Conclusions

The work presented here consists of a number of separate components which are combined to solve the larger problem of body surface reconstruction and the calculation of some single physical measurements such as body volume and surface area. Each component has its own strengths and shortcomings (each one discussed in the relevant section), and hence the success of the project is variable.

The complete method has a chained structure, in the sense that the output data from each component is the input data for the next one. Therefore, if any of the components is not completely effective, one cannot expect satisfactory results from any of the components that follow. This was found to be the case for the resampling procedures. Whenever the output of the resampling is such that helical topologies are introduced in the quadrilateral connectivity of samples, the result is ripples (artefacts) in the final surface. These artefacts are not a result of shortcomings in the surface interpolation method itself, which does not have this problem if the data set is not helical (see e.g. figures 31 and 32 in section 5.5, where the same interpolating procedure was used on hand-produced data that represent an ideally sampled set of curves), but are generated by the relative inadequacy of the proposed resampling techniques that take into account the topology of the objects but, in general, not their geometry.

This work has shown that in general, B-spline surfaces in conjunction with least square B-spline curves are very useful for skinning body scanner data as they implicitly deal with problems of noise, minor artefacts, contradictory scanner head data, poor sampling due to occlusion, smooth segment joining and branching topologies. The current implementation has proved to be fast and successful, and no unexpected problems were encountered. All methods and techniques proposed throughout this report were implemented unless explicitly stated otherwise.

Ultimately, this work demonstrates that the surface-from-point-set problem is greatly simplified if the body is first segmented and then curves are fitted along subsets of data in order to drive the actual surface patch fitting problem. It also makes clear that non-generic methods that incorporate assumptions that are related to the structure and geometry of the human body have a strong potential and are appropriate for expansion and further development.
APPENDICES
APPENDIX A : Bibliography - References

the density and hydration of fat-free mass’. Clinical Science, 1992, issue 82, pages 687-693.


APPENDIX B : Project planning and timeline

The problem of surface reconstruction from scanner data is quite open, it has therefore been rather difficult to produce a long term plan of action. The only known requirement in the beginning of the project was that the whole procedure had to be completely automatic, and that there had already been a significantly large piece of work (Least-squares B-spline curves fitting) for which code was already available and could potentially be used for building upon.

During the literature review, a decision had to be made concerning the general direction I should follow. After liaising with other members of the department, two things became clear:

- It would be wise to use the existing least-squares work, as it was already there and could solve most of the problems of the early stages of processing. Availability of existing code (for a PC/Windows platform) would save a lot of time and programming effort
- The implementation platform for all code produced should be Unix/Silicon Graphics (more details on that in appendix D)

At this stage, the problems would be broken down into the following sections:

- Pre-processing (segmentation, cleaning, LS curve fitting)
- Resampling
- Re-segmentation
- Surface fitting
- Smooth segment joining
- Volume and surface area calculation

The timeline initially produced is shown on page 86. However, delays were caused due to problems experienced during the design of the curve resampling process, as well as due to lack of time during the MRes taught module examinations (end of February - beginning of March). Only in middle march, when the taught component was over, did it become possible to start working intensively on the project, and sometime in May it became clear that it would not be possible to accomplish all project aims (specifically to produce a surface that covers the whole body), and the whole plan had to be reassessed.

The timeline of the project as it actually turned out is shown on page 87.
insert initial timeline here
insert final timeline here
APPENDIX C : Significance of the research

In its current form, the work has furthered the study of methods of skinning body data, as well as the methods for calculating body volume and surface area.

There are two main groups of beneficiaries of accurate computerised representations of human body forms and accurate volume/surface area measurements:

- **Clothing retailers and their users:** The availability of accurate body models could bring a revolution to the clothing industry, by enabling the rapid design and manufacturing of clothes that are guaranteed to fit properly. By taking advantage of the speed and accuracy of a computerised body modelling technique, it would be possible to acquire a set of measurements on a human body, upon which the rapid development of custom-made clothes can be based. These measurements are guaranteed to follow a unique standard, as opposed to the current situation where each tailor and designer uses his/her own empirical way of acquiring a measurement set. Moreover, the procedure of automatically creating a body model using a set of measurements is expected to be much faster than the current manual procedure which is both slow (typically 45-60 minutes per person) and inherently inaccurate since it is carried out by hand. An extra advantage is that development of a fast and fully automatic modelling/measuring procedure would facilitate a large scale sizing survey in order to redefine the cloth size ranges and scales, since it would make possible to measure in a reasonable amount of time a sample of people large enough to be statistically meaningful (approximately 20000 people). All the above would result in a significant reduction of the return rates of clothes purchases (currently about 20% for high-street retailers and 40% for mail order purchases), leading to increase of profit for the industry, and to significantly more satisfied customers.

- **Medical researchers:** There are certain areas of medical research where it would be desirable to have a computerized model of the patient’s body with which to work, rather than using the actual body of the patient itself. One such area is body growth monitoring, useful mainly in nutritional medicine for obesity and anorexia research. Acquiring successive scans of a subject at regular time intervals would make possible the detection of changes in body structure and enable study of how this can be affected by variations in types and dosages of nutritional elements (i.e. changes in diet). Another area is that of global measurements, such as body volume and surface area. For the latter (surface area), very simply there is so far no method of measuring it, if for a living person. The only available estimates for the area of a human skin have been acquired by skinning corpses and stretching pieces of the skin on a flat surface. As for body volume, the most dominant methods currently used are immersion tanks and
pressure pods. The first method consists of fully immersing a subject into a tank of water and then measuring the volume of displaced water. The second consists of placing the subject into an airtight pod, pumping in a known quantity of air, and then inferring the body volume from the difference in pressure. Both methods are time consuming, highly intrusive, almost guaranteed to cause discomfort to the subject, and simply impossible to carry out on certain categories of patients (e.g. frail, old and unwell patients). On the other hand, using a 12-second, non-contact body scanner would enable accurate measuring of body volume and surface area while causing minimal distress to the patient.
APPENDIX D: Platform issues

Due to time constraints, emphasis was given mostly on the design of a proper method that would meet the requirements associated with the definition of a precise and semantically correct surface that describes the human body, rather than generating a full application to be potentially used for commercial purposes. The code provided along with this report is therefore more of a prototype, developed and used in order to investigate and study the correctness, feasibility and performance of all methods and techniques described herein.

For the B-spline curve fitting, the code developed by Elliot West was used as provided. Only a few modifications were made (with his permission) in order to make results transferable and readable by the code developed for the purposes of this project. That code was developed on an Intel Pentium platform using Microsoft Visual C++ with MFC and the OpenGL libraries. Nevertheless, it has been decided at an early stage to develop code for this project in a UNIX platform, using the Silicon Graphics workstations. The reasons for making that decision were the following:

- Hardware requirements for such a project are rather high, and an appropriately powerful PC has not been available at the time when the project was carried out.
- On the other hand, Silicon Graphics workstations are readily available in the UCL CS department, especially for MRes CVPIGS students who have a whole lab equipped with such machines, on which they also have priority in using.
- Although it is a major requirement for the final application to run on a PC, this would only be useful if it could process a whole body within a few minutes. However, the code provided is quite slow in its current state of implementation (execution time is of the order of a couple of hours), and putting on the effort of speeding it up would be a time consuming task, beyond the scope of this project.

The solution initially chosen was to port the code provided and make it work on an SGI platform, but this proved to be extremely time consuming due to interface-dependent system calls. The solution that was finally chosen was to run the curve fitting code on the body scan (this takes about 2 hours but only has to be done once, and can even be left to run unattended overnight), and save the results (actually the control points of the fitted B-splines) for further use with the code developed (this would normally take about 2 minutes but has to be repeated since the code had been under constant development, therefore undergoing frequent changes).

This is not at all a convenient solution for a potential commercial user. Integration of the two pieces of code is a software engineering task for anyone interested in further development, should the methods proposed herein prove to be practically and academically acceptable.
APPENDIX E : User’s manual

The software was left while in mid-development due to time constraints and project aims. Therefore, the creation of a single application with a graphical user interface (according to the current market trends) was not completed. The software created for this project consists of four independent executable components (programs). However, these programs used together form a single skinning software package. They can be executed independently, since they communicate by using disk files. The user interface is strictly keyboard based. The use and functions of each of the four programs is discussed in more detail below.

**smpl - The body resampler**

This program processes body scan files that contain LS-fitted B-spline information and produces a sample set. To install it, please follow these steps:

- Copy the contents of the ‘smpl’ directory (supplied floppy disk) on an empty directory on a Silicon Graphics workstation.
- Type ‘gunzip sbbcpl06.blz’ and press Enter to decompress the supplied example scan file.
- Type ‘make’ and press Enter to compile the program (may take a few minutes).
- Type ‘smpl’ and press Enter to start the program.

Upon starting execution, the program will load and process the default file (sbbcpl06.blz). When processing is over, a window will open where the sampled model will be displayed. From this point on, the view can be manipulated by using the keyboard as described in the table below.

For all commands to work, the mouse pointer must be positioned within the display window. All commands that correspond to letters are case insensitive. However, for the arrow keys to work, the user must make sure that the NumLock key is ON.

<table>
<thead>
<tr>
<th>KEY</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Toggles control point mesh on/off</td>
</tr>
<tr>
<td>d</td>
<td>Toggles data point mesh on/off</td>
</tr>
<tr>
<td>Key</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>s</strong></td>
<td>Toggles slice curves on/off</td>
</tr>
<tr>
<td><strong>v</strong></td>
<td>Toggles key slice levels on/off. When on, a horizontal translucent plane is rendered for each slice, and indicates its location.</td>
</tr>
<tr>
<td><strong>j</strong></td>
<td>Toggles between single-slice mode and all-slices mode. When the program is in single slice mode, only the active slice is displayed. In all slices mode, slices at regular intervals along the body are displayed.</td>
</tr>
<tr>
<td><strong>o</strong></td>
<td>Moves one slice down (changes the active displayed slice - single-slice mode only)</td>
</tr>
<tr>
<td><strong>p</strong></td>
<td>Moves one slice up (changes the active displayed slice - single-slice mode only)</td>
</tr>
<tr>
<td><strong>e</strong></td>
<td>Increases slice jump step (all-slices mode only). This is to determine how many slices will be displayed. If jumpstep=1, all slices will be displayed, if jumpstep=2 then one every 2 slices is displayed, and so on. Every time e is pressed, the value of jumpstep increases</td>
</tr>
<tr>
<td><strong>a</strong></td>
<td>Toggles sample points on/off</td>
</tr>
<tr>
<td><strong>n</strong></td>
<td>Toggles curve normals on/off</td>
</tr>
<tr>
<td><strong>y</strong></td>
<td>Toggles lighting mode on/off. This works only when sample points are displayed. When lighting mode is on, samples are rendered using one color for all of them, otherwise they are coloured according to their location on the curve.</td>
</tr>
<tr>
<td><strong>q</strong></td>
<td>Toggles slice centroids on/off</td>
</tr>
<tr>
<td><strong>f</strong></td>
<td>Toggles frame of reference on/off</td>
</tr>
<tr>
<td><strong>z</strong></td>
<td>Toggles between orthogonal and perspective projection mode</td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>(centre of numeric keypad) Toggles between rotate mode and pan mode</td>
</tr>
<tr>
<td><strong>UP</strong></td>
<td>Rotates left around the X-axis (rotate mode) or moves up (pan mode)</td>
</tr>
<tr>
<td><strong>DOWN</strong></td>
<td>Rotates right around the X-axis (rotate mode) or moves down (pan mode)</td>
</tr>
<tr>
<td><strong>LEFT</strong></td>
<td>Rotates left around the Y-axis (rotate mode) or moves left (pan mode)</td>
</tr>
<tr>
<td><strong>RIGHT</strong></td>
<td>Rotates right around the Y-axis (rotate mode) or moves right (pan mode)</td>
</tr>
<tr>
<td><strong>0</strong></td>
<td>Rotates left around the Z-axis</td>
</tr>
<tr>
<td><strong>.</strong></td>
<td>Rotates right around the Z-axis</td>
</tr>
<tr>
<td><strong>+</strong></td>
<td>Zooms in</td>
</tr>
<tr>
<td><strong>-</strong></td>
<td>Zooms out</td>
</tr>
<tr>
<td><strong>]</strong></td>
<td>Increases the size of points</td>
</tr>
<tr>
<td><strong>[</strong></td>
<td>Decreases the size of points</td>
</tr>
</tbody>
</table>
Changes the colour mode. Color mode refers to the colour of the data points. Each time
the key is pressed, the colour mode changes in order to enhance different types of detail
in the point set (intensity, camera of origin, etc.)

Loads a file. When this key is pressed, a prompt will appear on the X-term window
where the program was started from. The user is required to position the mouse pointer
within this window and type the name of a *.bls file, with the pathname if necessary.
Valid files are all body scans that have been processed with Elliot West’s body.exe.
When the program starts, a default file (supplied with the program code) is loaded.

Reloads the current file.

Show a help screen

Exits the program

skin - The body skinner

This program processes *.bod files that are the output of the ‘smpl’ program. To install it, please
follow these steps:

- Copy the contents of the ‘skin’ directory (supplied floppy disk) on an empty directory on a Silicon
  Graphics workstation.
- Type ‘make’ and press Enter to compile the program (may take a few minutes).
- Type ‘skin’ and press Enter to start the program.

Upon starting execution, the program will load and process the default file (sbb06.bod). When
processing is over, a window will open where the model will be displayed. From this point on, the view
can be manipulated by using the keyboard as described in the table below.

For all commands to work, the mouse pointer must be positioned within the display window. All
commands that correspond to letters are case insensitive. However, for the arrow keys to work, the user
must make sure that the NumLock key is ON.
<table>
<thead>
<tr>
<th><strong>KEY</strong></th>
<th><strong>FUNCTION</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Toggles control point mesh on/off</td>
</tr>
<tr>
<td>d</td>
<td>Toggles data point mesh on/off</td>
</tr>
<tr>
<td>s</td>
<td>Toggles rendered surface on/off</td>
</tr>
<tr>
<td>v</td>
<td>Toggles interpolated curves on/off</td>
</tr>
<tr>
<td>o</td>
<td>Toggles horizontal interpolated curves on/off</td>
</tr>
<tr>
<td>p</td>
<td>Toggles vertical interpolated curves on/off</td>
</tr>
<tr>
<td>f</td>
<td>Toggles frame of reference on/off</td>
</tr>
<tr>
<td>t</td>
<td>Toggles triangulated mesh on/off</td>
</tr>
<tr>
<td>z</td>
<td>Toggles between orthogonal and perspective projection mode</td>
</tr>
<tr>
<td>5</td>
<td>(centre of numeric keypad) Toggles between rotate mode and pan mode</td>
</tr>
<tr>
<td>UP</td>
<td>Rotates left around X-axis (rotate mode) or moves up (pan mode)</td>
</tr>
<tr>
<td>DOWN</td>
<td>Rotates right around X-axis (rotate mode) or moves down (pan mode)</td>
</tr>
<tr>
<td>LEFT</td>
<td>Rotates left around Y-axis (rotate mode) or moves left (pan mode)</td>
</tr>
<tr>
<td>RIGHT</td>
<td>Rotates right around Y-axis (rotate mode) or moves right (pan mode)</td>
</tr>
<tr>
<td>0</td>
<td>Rotates left around Z-axis</td>
</tr>
<tr>
<td>.</td>
<td>Rotates right around Z-axis</td>
</tr>
<tr>
<td>+</td>
<td>Zooms in</td>
</tr>
<tr>
<td>-</td>
<td>Zooms out</td>
</tr>
<tr>
<td>l</td>
<td>Loads a file. When this key is pressed, a prompt will appear on the X-term window where the program was started from. The user is required to position the mouse pointer within this window and type the name of a *.bod file, with the pathname if necessary. Valid files are all body scans that have been processed with the “smpl” program. When the program starts, a default file (supplied with the program code) is loaded.</td>
</tr>
<tr>
<td>r</td>
<td>Reloads the current file</td>
</tr>
<tr>
<td>u</td>
<td>Reloads 'colours.ini' (the file with the color scheme). This allows the user to edit 'colours.ini’ while the program is running, then change display colours by pressing u.</td>
</tr>
<tr>
<td>Shift-#</td>
<td>(where # can be any number between 1 and 4) Toggles respective secondary segments on/off.</td>
</tr>
<tr>
<td>h</td>
<td>Shows a help screen</td>
</tr>
<tr>
<td>Esc</td>
<td>Exits the program</td>
</tr>
</tbody>
</table>

*triang - the B-spline enclosed area calculation utility using triangulation*
This is a sample program. It processes a set of data points by interpolating a B-spline curve along them and then calculating the enclosed area of the curve using triangulation. To install it, please follow these steps:

- Copy the contents of the ‘triang’ directory (supplied floppy disk) on an empty directory on a Silicon Graphics workstation.
- Type ‘make’ and press Enter to compile the program (may take a few minutes).
- Type ‘triang’ and press Enter to start the program.

Upon starting execution, a window will open where the sample curve will be displayed. From this point on, the program can be operated by using the keyboard as described in the table below.

For all commands to work, the mouse pointer must be positioned within the display window. All commands that correspond to letters are case insensitive.

<table>
<thead>
<tr>
<th>KEY</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Toggles control polygon on/off</td>
</tr>
<tr>
<td>d</td>
<td>Toggles data point mesh on/off</td>
</tr>
<tr>
<td>s</td>
<td>Toggles curve on/off</td>
</tr>
<tr>
<td>+</td>
<td>Increases the number of samples and recalculate enclosed area</td>
</tr>
<tr>
<td>_</td>
<td>Resets the to a minimum value (and recalculate enclosed area)</td>
</tr>
<tr>
<td>r</td>
<td>Iteratively increases the number of samples until the calculated value of enclosed area converges.</td>
</tr>
<tr>
<td>Esc</td>
<td>Exits the program</td>
</tr>
</tbody>
</table>

**simpson - the B-spline enclosed area calculation utility using Simpson’s rule**

This is a sample program. It processes a set of data points by interpolating a B-spline curve along them and then calculating the enclosed area of the curve using Simpson’s rule. To install it, please follow these steps:
• Copy the contents of the ‘simpson’ directory (supplied floppy disk) on an empty directory on a Silicon Graphics workstation.
• Type ‘make’ and press Enter to compile the program (may take a few minutes).
• Type ‘simpson’ and press Enter to start the program.

Upon starting execution, a window will open where the sample curve will be displayed. From this point on, the program can be operated in exactly the same way as the ‘triang’ program. All comments and keyboard commands as described above for the case of ‘triang’ apply here without any change.
APPENDIX F: Program code

(Please see attached floppy disk)
# Table of contents

## 1. Summary of the problem area

1.1 Background  
1.2 Motivation  
1.3 Scanning technologies  
1.4 The Hamamatsu Body Lines Scanner  
1.5 Scope of the project  
1.6 Contributions  
1.7 Problem Description

## 2. Literature Review

2.1 Overview of previous work  
2.2 LASS  
2.3 Hughes Hoppe  
2.4 Stoddart and Hilton  
2.5 Previous MRes project work  
2.6 Body Scanner work at UCL  
2.7 Other body modelling work in progress at UCL

## 3. Design and Methodology

3.1 Overview of the proposed algorithm  
3.2 Problems  
3.3 Assumptions  
3.4 Scanning procedure and data clipping  
3.5 Least squares curve fitting (overview)  
3.6 Body resegmentation

## 4. Curve resampling

4.1 Uniform parametric intervals  
4.2 Torso slice growing based on closest distance from previous sample  
4.3 Torso slice growing based on intersection with curve normal  
4.4 Torso slice growing based on intersection of cord bisector  
4.5 Mid-bust refining  
4.6 Leg slice growing based on bounding box  
4.7 Leg slice growing based on lateral plane intersection
5. Surface fitting

5.1 B-spline curve interpolation
5.2 B-spline surface interpolation
5.3 Ensuring smooth continuity between successive cubic B-Splines during interpolation.
5.4 B-spline surface fitting on a smoothly closed deformed cylinder
5.5 Smooth segment joining

6. Volume and surface area calculation

6.1 Volume calculation
   6.1.1 Calculation of the enclosed area in a closed B-spline curve.
   6.1.2 Volume calculation by integrating slice areas
6.2 Surface area calculation
   6.2.1 Slice length calculation
   6.2.2 Surface area calculation by integrating slice lengths
   6.2.3 Alternative proposal: Surface area calculation by recursive subdivision of B-spline surfaces

7. Results

7.1 Output of the algorithm
7.2 Analysis and critique of the results

8. Conclusions

8.1 Further work
8.2 Final Conclusions

APPENDIX A : Bibliography - References
APPENDIX B : Project planning and timeline
APPENDIX C : Significance of the research
APPENDIX D : Platform issues
APPENDIX E : User’s manual
APPENDIX F : Program code