

Registration of the  
Human Femur  
by  
Christopher Bridge (PEM)

Fourth-year undergraduate project  
in Group F, 2012/2013

I hereby declare that, except where specifically indicated, the work submitted herein is my own original work.

Signed: \_\_\_\_\_ Date: \_\_\_\_\_

# Registration of the Human Femur

by Christopher Bridge (PEM)

Group F 2012/2013

## Technical Abstract

A research aim within the Medical Imaging Group at Cambridge University Engineering Department is to develop a methodology for measuring thickness of femoral *cortical bone* and comparing the results across large numbers of patients. This has potential implications for the study of hip fracture, since cortical thickness is thought to be a key determinant of a patient's susceptibility to fracture.

Once the femoral surface and cortical thickness have been extracted from computed tomography (CT) data, it is necessary to *register* the surfaces to some *canonical model* of the femur. This allows further comparison of many different patients in a standard space despite potentially large variations in the shape of their femurs. The current surface registration process uses a global affine stage followed by a local B-spline stage inside an iterative closest point (ICP) optimisation framework. This project aimed to evaluate and improve various aspects of this registration process.

A number of *registration failures* are observed using the existing process, where key anatomical features on the two surfaces (in particular the lesser trochanters) are not brought into correspondence. A technique of extracting *distinguished points* (specifically the tips of the greater and lesser trochanters) and incorporating these into the ICP cost function was found to solve the majority of the registration failures in a dataset of over 600 femurs. In order to locate these distinguished points successfully, it is necessary to consider the *differential geometry* of the surface including both first order and second order information in the form of surface normals and Gaussian and mean curvature estimates. The drawback of this method is that when the distinguished point detection fails (as it was found to do on a very small number of unusual surfaces), the resulting registration is meaningless.

A second problem observed with the existing registration process is unwanted deformation or 'warping' of the surface (particularly around the shaft), which is not necessary to register the surfaces. In order to quantify this, a 'warping metric' was developed based on the transformed position of points that lie in approximately the same plane on the undeformed surface. The rationale being that these points should remain approximately coplanar after the transformation, so measuring the distance of points from the plane before and after registration gives a numerical measure of warping. This metric is simplistic, but gives some insight into the problem and allows quantitative comparison between different registration processes.

In an attempt to find a more *localised* transformation model to reduce warping, a *locally affine* transformation was implemented and tested. This is a non-parametric algorithm that works by finding rigid-body transformations to bring small patches of the surfaces into alignment. These transformations are then smoothed over the surface to enforce smoothness of the overall transformation, resulting in a different affine transformation at each point. The algorithm was found to compare favourably to the B-spline transformation in terms of the warping undergone by the surfaces during registration.

Alternative *point matching* schemes were introduced with the dual aims of reducing registration failures and reducing warping. These alter the simple ‘closest point’ criterion used in the standard ICP framework. In one such scheme, some weight is given to the similarity of the surface normals at the two points as well as the the distance between them in 3D space (the *normal-weighted* method). This scheme was found to be useful in reducing registration failures, but in fact introduced additional warping to the registrations. A second scheme favours points that lie along the normal direction of the canonical surface (the so-called *normal-shooting* method). This was found to give have slight advantages in terms of both registration failures and warping. A third alteration to the original scheme involves ignoring point matches where one point lies on a rim of the target surface. This helped solve registration failures in situations where there are regions of the canonical surface that do not overlap with a region of the target surface.

It was thought that recently developed matching techniques borrowed from the field of Computer Graphics may be able to model the variation in head-neck angle seen across femurs, as this is similar to the *articulated motion* that such algorithms were developed for. One such scheme that was found and tested involved *conformal flattening* of the surfaces to the complex plane, followed by an isometric Möbius transformation within the complex plane to register the two flattened representations. Initial tests suggested that the scheme is computationally very expensive and does not reduce warping as desired.

*Synthetic surfaces* with known *ground truth* correspondences were used as a further tool for evaluating and investigating registration processes. These tests revealed that none of the methods discussed were able to come close the the ‘best’ transformation when the correspondences were known, but that using *normal-weighted* point matching generally came closer than the standard closest point matching. It also showed that ICP methods are very susceptible to problems with local minima, sometimes even in the global alignment stage, and that consequently the performance of the optimisation depends heavily on initialisation.

The most important contributions of this project are the successful distinguished point scheme for reducing registration failures and the use of the locally affine transformation to reduce warping. The other findings of the report may be useful to those developing registration algorithms for femurs or similar bone surfaces.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Surface Registration . . . . .	2
1.3	Existing Registration Process . . . . .	2
1.4	Report Outline . . . . .	4
1.5	Implementation Details . . . . .	4
1.6	Notation and Abbreviations . . . . .	5
<b>2</b>	<b>Distinguished Point Detection</b>	<b>6</b>
2.1	Registration Failures . . . . .	6
2.2	Previous Work . . . . .	7
2.3	Distinguished Point Detection Algorithm . . . . .	8
2.3.1	Estimating Surface Curvatures . . . . .	9
2.3.2	Choosing Distinguished Points . . . . .	11
<b>3</b>	<b>Evaluating Registration Quality</b>	<b>14</b>
3.1	Previous Work . . . . .	15
3.2	Mesh Distortion Metric . . . . .	15
3.3	Synthetic Data and Ground Truth Error . . . . .	16
<b>4</b>	<b>Alternative Transformations</b>	<b>18</b>
4.1	Previous Work . . . . .	19
4.2	Constrained (‘Physical’) Transformations . . . . .	19
4.2.1	Global ‘Physical’ Transformation . . . . .	20
4.2.2	‘Physical’ B-spline Transformation . . . . .	20
4.3	Locally Affine Deformation . . . . .	21
<b>5</b>	<b>Alternative Methods for Finding Matched Points</b>	<b>23</b>
5.1	Previous Work . . . . .	23
5.2	Normal Transformations . . . . .	26
5.3	Normal Weighted Point Matching . . . . .	26
5.4	Normal ‘Shooting’ Point Matching . . . . .	27
5.5	Ignoring Rim Point Matches . . . . .	27
5.6	Conformal Flattening Method . . . . .	28
<b>6</b>	<b>Results</b>	<b>30</b>
6.1	Registration Failures . . . . .	30
6.2	Mesh Distortion . . . . .	30
6.3	Alternative Matching Methods . . . . .	32

6.4	Synthetic Data . . . . .	32
6.5	Conformal Flattening . . . . .	34
<b>7</b>	<b>Discussion</b>	<b>35</b>
7.1	Distinguished Points . . . . .	35
7.2	Alternative Transformations . . . . .	36
7.3	Alternative Matching Schemes . . . . .	37
7.4	Conformal Flattening Method . . . . .	38
7.5	Distortion Metric . . . . .	39
7.6	Synthetic Data and Local Minima . . . . .	40
<b>8</b>	<b>Conclusions</b>	<b>41</b>
<b>A</b>	<b>Risk Assessment Retrospective</b>	<b>I</b>

# 1 Introduction

## 1.1 Motivation

Hip fracture, defined as fracture of the proximal femur, is a common and serious medical condition leading to severe debilitation and a 33% mortality rate at one year [1]. Care for hip fracture sufferers costs the United Kingdom an estimated £2 billion a year [2], and as such it is an important medical concern especially within an ageing population.

The anatomy of the proximal femur is outlined in Figure 1a, and is sufficiently detailed for the purposes of this report.

Research has suggested that the thickness of the *cortical layer* of bone (the outer, dense layer) is a key determinant in a person's susceptibility to hip fracture [3]. Current work is aiming to conduct *cohort studies* to compare features of the cortical thickness distributions across large numbers of patients and controls. This will potentially assist future studies into the condition and its causes, allow clinicians to identify those patients particularly at risk [4], and help in the development of preventative medicines by providing a method for evaluating treatment [5].

A research project within the Medical Imaging Group at Cambridge University Engineering Department has already demonstrated a technique for estimating cortical thickness from computed tomography (CT) data [6]. This allows a *cortical thickness map* over the surface of a femur (such as that in Figure 1b) to be extracted from an *in vivo* scan.

However, before any meaningful comparisons can be drawn between cortical thickness maps obtained from different patients, it is necessary to *register* the surfaces to some standard (or *canonical*) shape. After a successful registration, an individual's cortical thickness values can be mapped onto the canonical model, and all further statistical comparison between sets of thickness maps can take place on this canonical surface. The goals of designing a registration process are therefore to obtain anatomically meaningful correspondences, and to be highly automated to allow large cohort studies to be carried

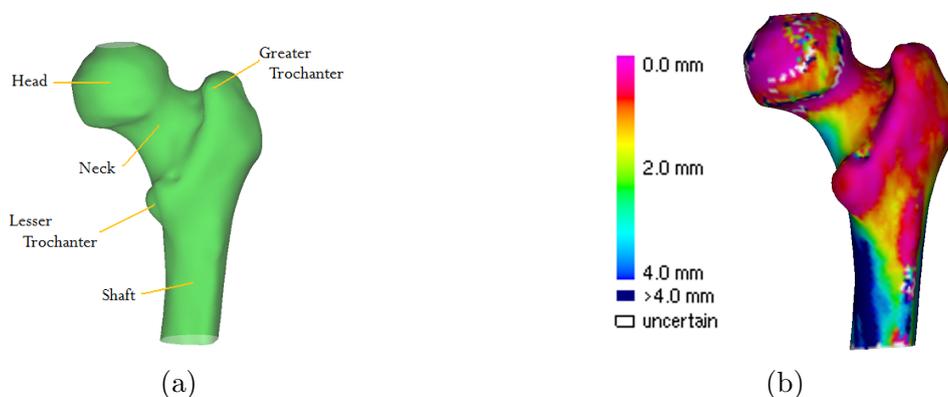


Figure 1: (a) Anatomy of the human femur (proximal end). (b) A cortical thickness map for a 3D femur surface.

out in an efficient and repeatable manner. This project is concerned with improving the existing surface registration methodology towards these goals.

## 1.2 Surface Registration

In this report, we consider a *surface* to be composed of a set of vertices in 3D space, and a set of triangles connecting them. Surface  $\mathcal{S}_A$ , the canonical femur, consists of the set of vertices  $\{\mathbf{p}_i\}$  and surface normals  $\{\mathbf{n}_i\}$ , and surface  $\mathcal{S}_B$ , the individual ('target') femur in question, consists of the set of vertices  $\{\mathbf{q}_i\}$  and surface normals  $\{\mathbf{r}_i\}$ . Surface registration between two surfaces  $\mathcal{S}_A$  and  $\mathcal{S}_B$  is then *the estimation of a mapping between coordinate systems  $Ref_A$  and  $Ref_B$*  such that anatomically corresponding points are brought into correspondence (as defined by Audette et al. [7]). Usually the process involves finding a transformation applied to surface  $\mathcal{S}_A$  such that it is optimally aligned with surface  $\mathcal{S}_B$  by some measure. At the end of our registration process, we wish to have, for each vertex of surface  $\mathcal{S}_A$ , a corresponding point on surface  $\mathcal{S}_B$  via the transformation such that we can map across cortical thickness data.

There are three key decisions to be made when designing a registration process:

1. **Transformation** – The form and parametrisation of the transformation must be chosen. This implicitly defines the degrees of freedom of the transformation and has implications for the sort of variation that can be modelled.
2. **Similarity Criterion** – Some measure of similarity must be used to determine the 'best' transformation. In iterative optimisation schemes, this is defined by the choice of *cost function*.
3. **Optimisation Strategy** – In some cases, there may be a closed-form solution, but usually some form of optimisation scheme will be necessary.

There is a large body of literature describing methods to perform surface registration, and a diverse range of approaches have been taken by the researchers, especially for *non-rigid* registration as required here (Audette et al. give a thorough overview [7]). The task in our particular case is complicated by the fact that the surfaces to be registered come from different patients, whose femurs can vary widely in shape. It is therefore often not obvious which points should be placed in correspondence.

## 1.3 Existing Registration Process

The *wxRegSurf* software currently used by the group for registration makes use of an iterative registration framework called the *iterative closest point* (ICP) method, as introduced by Besl and McKay [8], and Chen and Medioni [9]. This involves two steps that are repeated until convergence or termination:

1. **Matching Stage** – For each point on surface  $\mathcal{S}_A$  find the closest point on surface  $\mathcal{S}_B$ . Alternatively point pairs can be found based on some other criterion.
2. **Transforming Stage** – Find the transformation such that the sum of the distances between matched points is minimised. In the existing implementation this is done using the *Levenberg-Marquardt algorithm*, a standard non-linear least-squares optimisation routine, to find the optimal parameters.

The full registration process takes place in three stages, as shown in Figure 2:

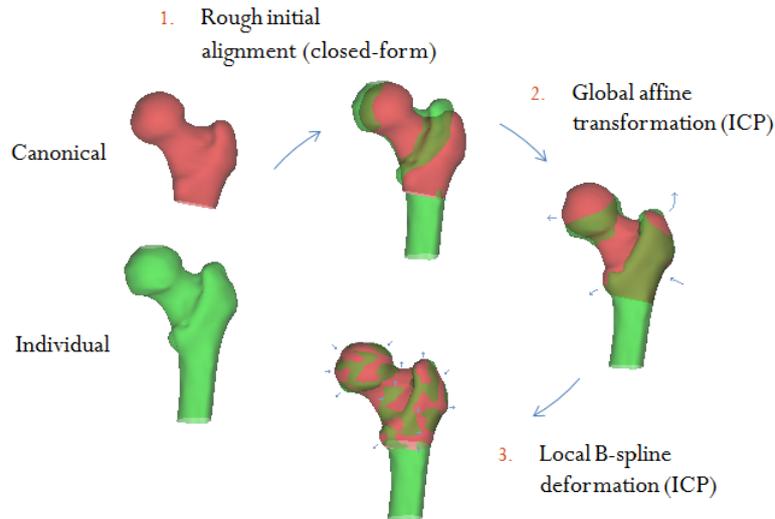


Figure 2: The three stages of the existing registration process.

1. **Rigid-body rough alignment**, based on aligning the estimated *natural axes* of the surfaces: one aligned with the shaft (the proximal-distal direction), one with the neck (the lateral direction), and a third perpendicular to these (the posterior-anterior direction). This is a closed-form process.
2. **Global affine transformation**, where the 12 transformation coefficients are found using the ICP method.
3. **Local B-spline deformation**, based on interpolation from a grid of deformable control points, as used by Rueckert et al [10]. The control point positions are determined using the ICP method.

Use of the ICP framework means that there are two choices that govern the effective *similarity criterion*: the method for choosing *closest points*, and the *cost function* for the optimisation of the transformation. The effective similarity criterion of the above methodology is the total distance between closest points on the surfaces.

## 1.4 Report Outline

This report details investigations into various improvements to the current registration process:

- Sometimes the existing method fails to bring anatomically corresponding points (such as the trochanters) into alignment. In Section 2, a method for the extraction and use of *distinguished points* is given. This represents a change to the similarity criterion by changing the cost function of the optimisation process.
- The conclusions drawn from comparisons of thickness maps are only useful if the registration method does not unrealistically distort the mesh. In Section 3, a simple metric is developed to quantify the distortion undergone by a mesh.
- In Section 4, *alternative transformations* to minimise distortion of the mesh are introduced.
- As an alternative solution, Section 5 describes ways to alter the similarity criterion by using different ways to select ‘closest’ points within the ICP framework.
- The results of the experiments conducted are presented in Section 6.
- The discussion of the results is presented in Section 7, and the conclusions are presented in Section 8.

This report does not consider changes to the optimisation strategy itself.

## 1.5 Implementation Details

The *wxRegSurf* software is written in C++ using the cross-platform GUI library *wxWidgets*. Unless otherwise stated, all experiments were conducted by editing and adding to the existing C++ code.

## 1.6 Notation and Abbreviations

The following mathematical notation will be used throughout this report:

$i, j$	$\in \mathbb{N}_0$	A variable denoting the <i>index</i> of a <i>vertex</i> in a list
$k$	$\in \mathbb{N}_0$	A variable denoting the <i>index</i> of a <i>triangle</i> in a list
$\mathcal{S}_A$		Surface $A$ , representing the canonical femur
$\mathcal{S}_B$		Surface $B$ , representing the target femur
$t$	$\in \mathbb{N}_0$	Time index in an iterative process (written in superscript)
$\mathbf{p}_i$	$\in \mathbb{R}^3$	The <i>position vector</i> of the vertex $i$ on $\mathcal{S}_A$
$\mathbf{q}_i$	$\in \mathbb{R}^3$	The <i>position vector</i> of the vertex $i$ on $\mathcal{S}_B$
$\mathbf{p}_{\text{src}}$	$\in \mathbb{R}^3$	The position vector of the <i>shaft rim centre</i> of $\mathcal{S}_A$
$\mathbf{p}_{\text{cm}}$	$\in \mathbb{R}^3$	The position vector of the estimated <i>centre of mass</i> of $\mathcal{S}_A$
$\mathbf{n}_i$	$\in \mathbb{R}^3$	The unit, outward-facing <i>surface normal vector</i> at vertex $i$ on $\mathcal{S}_A$
$\mathbf{r}_i$	$\in \mathbb{R}^3$	The unit, outward-facing <i>surface normal vector</i> at vertex $i$ on $\mathcal{S}_B$
$\mathbf{m}_k$	$\in \mathbb{R}^3$	The unit, outward-facing <i>surface normal vector</i> at triangle $k$
$\hat{\mathbf{e}}_s, \hat{\mathbf{e}}_n, \hat{\mathbf{e}}_o$	$\in \mathbb{R}^3$	Orthogonal unit vectors of the <i>natural axes</i> system in the <i>shaft</i> , <i>neck</i> and <i>other</i> directions respectively
$K_i$	$\in \mathbb{R}$	Estimated <i>Gaussian curvature</i> at vertex $i$
$H_i$	$\in \mathbb{R}$	Estimated <i>mean curvature</i> at vertex $i$
$\mathcal{T}(\cdot)$	$: \mathbb{R}^3 \rightarrow \mathbb{R}^3$	A general description of a transformation applied to vertices
$\mathcal{N}_{\mathcal{T}}(\cdot)$	$: \mathbb{R}^3 \rightarrow \mathbb{R}^3$	The transformation applied to a surface's normals when $\mathcal{T}$ is applied to its vertices

The following abbreviations are used:

3D	3-dimensional
DoF	Degree(s) of freedom
FFD	Free-form deformation
ICP	Iterative Closest Point, a registration framework
LAD	Locally Affine Deformation

## 2 Distinguished Point Detection

In this section, the problem of *registration failures* in the existing registration process is considered, and ways to reduce their occurrence are explored. A *distinguished point* detection algorithm for this purpose is presented.

### 2.1 Registration Failures

One limitation of the current software is the nature of the standard ICP cost function, which simply minimises the total distance between closest points on the two surfaces. If the initial alignment is sensible and the two surfaces are not too dissimilar in shape, the current technique works well and *anatomically corresponding* points end up being aligned in the final registration. However the process does not *directly* ensure that this is the case as it takes no explicit account of the *shape* of the surfaces. This is illustrated by the extreme example shown in Figure 3, in which it can be seen that, although the two surfaces are closely aligned, the registration process has completely failed to align the two lesser trochanters. Such *registration failures* must be manually identified and corrected, a time-consuming, subjective, and non-repeatable process.

Two types of registration failures have been observed. The first is when the alignment is completely wrong and the registration process has failed to align the anatomical features properly. We will refer to such failures as *complete failures*. These can be identified from visualisation with little room for ambiguity. The second, more common type of failure is when most features of the surfaces are aligned, but the lesser trochanters are incorrectly aligned as in Figure 3. We will refer to such failures (where the lesser trochanters overlap by less than 2/3 their height or width) as *lesser trochanter failures*.

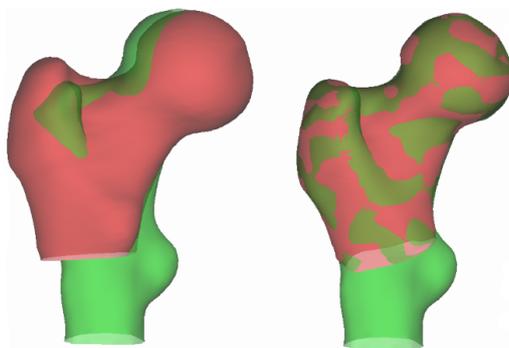


Figure 3: Example of a registration failure. *Left* the red canonical femur and the green individual femur after the initial alignment, and *right* the registered femurs, showing the failure to align the lesser trochanters.

## 2.2 Previous Work

Many researchers have considered the use of the *differential geometry* of a surface as a way of quantitatively describing its *local shape*. The first derivative of a surface at a point may be described by the (outward-facing) *surface normal vector* at that point. Whilst this is useful, a richer description can be obtained by also considering the second-order information, characterised by *surface curvatures*. Besl and Jain [11] and Brady et al. [12] provide detailed overviews of the use of surface curvatures.

Consider a point  $\mathbf{p}$  on a smooth surface in  $\mathbb{R}^3$ , with an outward facing surface normal  $\mathbf{n}$ , as shown in Figure 4. For each direction within the surface at  $\mathbf{p}$ , one can define a plane  $A$  that intersects the surface in that direction, and also contains the surface normal  $\mathbf{n}$ . The intersection of this plane with the surface forms a plane curve,  $C$ . The *surface curvature* at  $\mathbf{p}$  in this direction is defined as the curvature of this plane curve at  $\mathbf{p}$ . Curvature in the other directions may be obtained by rotating the plane  $A$  around  $\mathbf{n}$ . For a smooth surface there will be two directions of extreme curvature termed the *principal directions*, which are perpendicular. The curvature values in these directions (the maximum and minimum curvatures) are the *principal curvatures*,  $\kappa_1$  and  $\kappa_2$ .

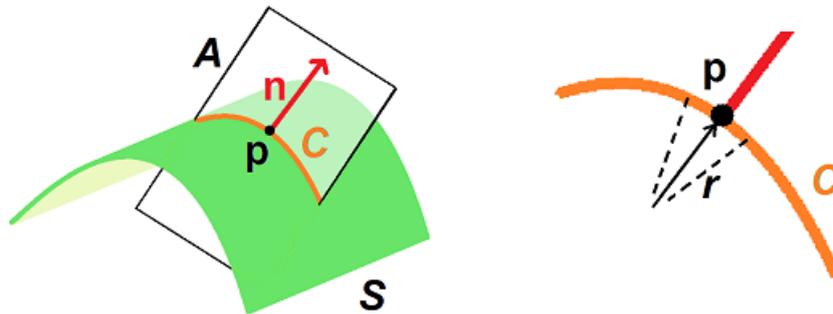


Figure 4: Definition of surface curvature.

Two further commonly used quantities are the *Gaussian curvature* ( $K$ ) and *mean curvature* ( $H$ ), defined as the product and mean of the principal curvatures as follows:

$$\begin{aligned} K &= \kappa_1 \kappa_2 \\ H &= \frac{\kappa_1 + \kappa_2}{2} \end{aligned}$$

The sign combination of the Gaussian and mean curvatures categorises the local shape as flat, ridge, valley, peak, pit etc. [11].

The general approach to preventing registration failures involves the extraction of some sort of feature from the surfaces, usually using differential geometry, and the incorporation of these features into the similarity criterion to guide the registration process. Approaches broadly belong to three categories: extraction of *point*, *line*, and *region* features.

*Region* features are entire areas of a surface with similar characteristics (for example the sign combination of mean and Gaussian curvatures) or separated by some meaningful boundary. One can imagine, for example, identifying which vertices belong to the head, neck, shaft, and greater and lesser trochanters and using this information to help register the surfaces. Many techniques have arisen in the field of Computer Graphics for segmenting meshes into meaningful regions. However, many are designed to find fairly uniform patches separated by sharp edges or “watersheds” [13], and as such would not be suitable for the subtly curving shape of a femur. Other schemes do not have such limitations (such as those in Besl and Jain [14] and Yamauchi et al. [15]), but it is unclear whether they would produce useful results from a smooth surface such as a femur. Furthermore, region growing algorithms are involved and computationally expensive, and classification of every point is somewhat excessive for our purposes here.

*Line* features, such as *crest lines* (having various definitions related to principal curvatures [7, 16]), are useful for certain types of surface where there are many obvious crest lines such as the brain [17], but are of limited use for femurs, where the exact location of the lines would likely be determined largely by uninteresting small-scale features or noise in the mesh.

*Point* features are more suited to reducing registration failures in femurs. They generally involve simpler detection algorithms, and, once detected, may be straightforwardly incorporated into the existing ICP registration framework. Criteria for the selection of points generally involve extrema of geometric properties (such as principal curvatures [18] or Gaussian curvatures [19]). Once detected, *descriptors* of the point may be constructed (using, for example, histograms of neighbouring vertex properties [20] or the shape of the surface at a fixed radius from the point [21]), and these may then be used to find correspondences between the detected points.

## 2.3 Distinguished Point Detection Algorithm

The algorithm presented here uses information about the position, normal and curvature of the surface’s vertices to locate two very specific distinguished points: the tips of the greater and lesser trochanters. This has advantages over a general point detection scheme as it allows us to use our specific knowledge about the shape of femurs and the important features to match. It also avoids the need for a matching stage, which would add complication and may give spurious results. However it does mean that the algorithm is highly specialised to femurs and could not be applied to other surfaces.

The existing software already estimates the *centre of mass*,  $\mathbf{p}_{\text{cm}}$ , (average position of all vertices) and the *natural axes* of the two femurs (represented by the unit vectors  $\hat{\mathbf{e}}_s$ ,  $\hat{\mathbf{e}}_n$  and  $\hat{\mathbf{e}}_o$  respectively in the *shaft*, *neck* and *other* directions) for the purposes of performing the initial alignment (see Figure 5). This coordinate system, with the centre of mass as

the origin, gives a convenient representation to use when searching for the distinguished points. Additionally, the location of the *shaft rim centre*,  $\mathbf{p}_{\text{src}}$ , (the centre of the rim that represents the edge of the mesh around the shaft) is calculated by the existing software and used by the algorithm.

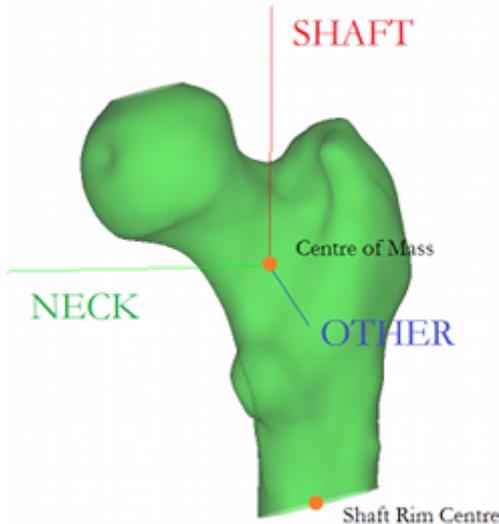


Figure 5: The natural axis coordinate system.

### 2.3.1 Estimating Surface Curvatures

Methods for estimating Gaussian and mean curvature values at points on an irregular mesh were adapted from Csákány and Wallace [22]. Methods for calculating more sophisticated estimates (as well as further differential properties such as principal directions) may be found in Taubin [23] and Meyer et al. [24]. However Csákány and Wallace’s scheme is simple to implement and sufficient for our purposes. The Gaussian curvature  $K_i$  at a vertex  $i$  is calculated from the *angle deficit* of those triangles *incident upon* vertex  $i$  (i.e. those triangles that have vertex  $i$  as one of their vertices). The mean curvature  $H_i$  is found by considering the normals of the same triangles. Note that the following notation relates to  $\mathcal{S}_A$ , but in practice the process is also performed on  $\mathcal{S}_B$  in an identical way.

We shall denote the set of triangular faces incident upon vertex  $i$  as  $\mathcal{F}_i$ , the side lengths of these triangles  $k$  as  $a_k$ ,  $b_k$  and  $c_k$  (where  $c_k$  is the side opposite vertex  $i$ ), their outward-facing unit normals as  $\mathbf{m}_k$ , and their areas as  $A_k$ . The points that share triangles with  $i$  are denoted  $\mathbf{p}_{i(k)}$  as defined in Figure 6. The triangles are ordered cyclically around  $i$  such that triangle  $(k + 1)$  is always the next triangle located anticlockwise from triangle  $k$  around vertex  $i$  when the surface is viewed from the outside, and it is the side of length  $b_k$  that is shared by the triangles  $k$  and  $(k + 1)$ .

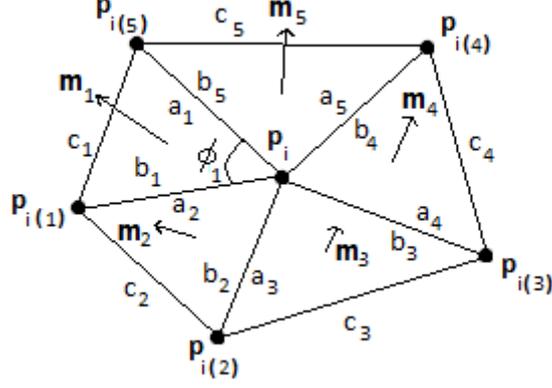


Figure 6: Definitions for the curvature calculations for five incident triangles.

The angle deficit  $\Delta_i$  at vertex  $i$  is then:

$$\Delta_i = 2\pi - \sum_{k \in \mathcal{F}_i} \phi_k$$

where  $\phi_k$  is the interior angle of triangle  $k$  where it meets vertex  $i$ , calculated using

$$\phi_k = \arccos \left( \frac{a_k^2 + b_k^2 - c_k^2}{2a_k b_k} \right)$$

Using this, the Gaussian curvature,  $K_i$ , and the mean curvature,  $H_i$ , are given by

$$K_i = \frac{2\Delta_i}{\sum_{k \in \mathcal{F}_i} A_k}$$

$$H_i = \frac{\sum_{k \in \mathcal{F}_i} \left( z_k b_k \arccos(\mathbf{m}_k \cdot \mathbf{m}_{k+1}) \right)}{2 \sum_{k \in \mathcal{F}_i} A_k}$$

where  $z_k \in \{-1, 1\}$  determines whether the surface bends in a concave or convex manner between triangles  $k$  and  $(k+1)$  as follows ( $\text{sgn}(\cdot)$  is the *signum function*):

$$z_k = \text{sgn} \left( (\mathbf{m}_k \times \mathbf{m}_{k+1}) \cdot (\mathbf{p}_{i(k)} - \mathbf{p}_i) \right)$$

The values of the mean and Gaussian curvature can be seen in Figure 7a. There is substantial noise reducing the usefulness of these estimates due to measurement error and the upstream segmentation method. Therefore a median filter is applied to the curvature values to eliminate outliers, followed by several passes of a mean filter to leave only the large-scale features: see Figure 7b. It is evident that the tips of the greater and lesser trochanters are among the regions of high Gaussian and mean curvature.

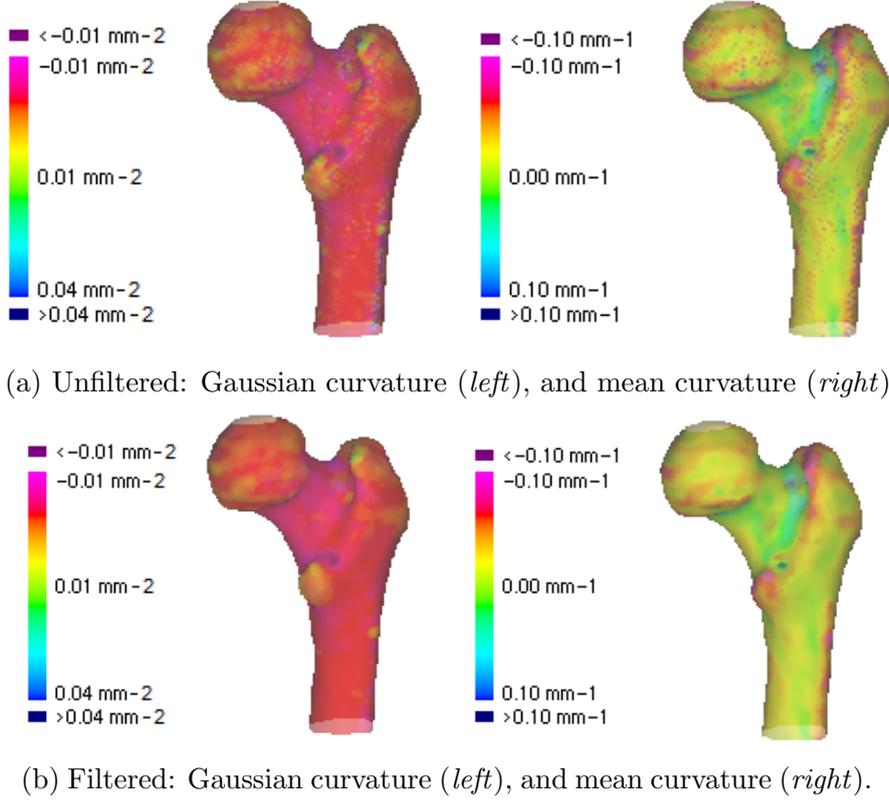


Figure 7: Visualisation of estimated surface curvatures.

### 2.3.2 Choosing Distinguished Points

Equipped with the estimated curvature values, the distinguished points algorithm starts by deciding whether the surface represents a left or right femur by summing the positive Gaussian curvature values at vertices on either ‘side’ of the femur (with respect to the other axis) below the centre of mass. The side with the larger sum is labelled as the side which contains the lesser trochanter (the posterior side).

The *lesser trochanter vertex* is then chosen to be the vertex  $i$  that satisfies all the following criteria. The point must:

1. Be on the posterior side of the centre of mass (with respect to the *other* axis) ( $((\mathbf{p}_i - \mathbf{p}_{\text{cm}}) \cdot \hat{\mathbf{e}}_o > 0)$ ), below the centre of mass with respect to the *shaft* axis ( $((\mathbf{p}_i - \mathbf{p}_{\text{cm}}) \cdot \hat{\mathbf{e}}_s < 0)$ ) and forward of the shaft rim centre with respect to the *neck* axis ( $((\mathbf{p}_i - \mathbf{p}_{\text{src}}) \cdot \hat{\mathbf{e}}_n > 0)$ ).

In cases where the surface contains a long section of the femoral shaft, the centre of mass is not a suitable threshold in the *shaft* direction. Therefore, in cases where the distance from ‘highest’ point on the surface (in the *shaft* direction),  $\mathbf{p}_h$ , to the centre of mass is greater than the furthest distance of any point from the centre of mass in *neck-other* plane (call this  $d_n$ ), an alternative criterion is used:

$$\|(\mathbf{p}_i - \mathbf{p}_h) \cdot \hat{\mathbf{e}}_s\| > d_n$$

2. Have a unit normal whose dot product with the unit *shaft* axis vector has an absolute value of less than 0.2 ( $\|\mathbf{n}_i \cdot \hat{\mathbf{e}}_s\| < 0.2$ ).
3. Have a positive value of Gaussian curvature ( $K_i > 0$ ).
4. Have a greater value of the following function than any other vertex satisfying the other criteria:

$$C_{i,LT} = \|(\mathbf{p}_i - \mathbf{p}_{src}) - ((\mathbf{p}_i - \mathbf{p}_{src}) \cdot \hat{\mathbf{e}}_s)\hat{\mathbf{e}}_s\|^2 + \lambda_{K,LT}K_i - \lambda_{H,LT}H_i$$

The first term in this measures the distance of the point from the shaft rim centre in the *neck-other* plane. Added to this are the Gaussian and mean curvatures weighted by the non-negative weighting factors  $\lambda_{K,LT}$  and  $\lambda_{H,LT}$ .

The *greater trochanter vertex* is chosen to be the vertex  $i$  that satisfies the following criteria. The point must:

1. Be located ‘behind’ the centre of mass with respect to the *neck* axis ( $((\mathbf{p}_i - \mathbf{p}_{cm}) \cdot \hat{\mathbf{e}}_n < 0)$ ), ‘above’ the centre of mass with respect to the *shaft* axis ( $((\mathbf{p}_i - \mathbf{p}_{cm}) \cdot \hat{\mathbf{e}}_s > 0)$ ) and on the posterior side of the centre of mass with respect to the *other* axis ( $((\mathbf{p}_i - \mathbf{p}_{cm}) \cdot \hat{\mathbf{e}}_o > 0)$ ).
2. Have a surface normal whose dot product with the increasing *neck* axis is greater than zero ( $\mathbf{n}_i \cdot \hat{\mathbf{e}}_n > 0$ ).
3. Have a greater value of the following function than any other function satisfying the other criteria:

$$C_{i,GT} = (\mathbf{p}_i - \mathbf{p}_{cm}) \cdot \hat{\mathbf{d}} + \lambda_n(\mathbf{n}_i \cdot \hat{\mathbf{d}}) + \lambda_{K,GT}K_i - \lambda_{H,GT}H_i$$

where  $\lambda_{K,GT}$ ,  $\lambda_{H,GT}$  and  $\lambda_n$  are non-negative weighting factors assigning variable weight to the Gaussian curvature, mean curvature and normal direction respectively. The unit vector  $\hat{\mathbf{d}}$  is the ‘desired direction’ for the position and normal direction of the greater trochanter, given by:

$$\hat{\mathbf{d}} = \frac{\hat{\mathbf{e}}_s + \hat{\mathbf{e}}_n + \hat{\mathbf{e}}_o}{\|\hat{\mathbf{e}}_s + \hat{\mathbf{e}}_n + \hat{\mathbf{e}}_o\|}$$

An example of the distinguished points selected by this scheme is shown in Figure 8. The scheme has a tendency to pick a lesser trochanter point away from the centre of the trochanter and towards the edges. This is because there is in general higher curvature towards the edges of the trochanter, and this scheme gives weight to the curvature value. To correct for this, once a point on the lesser trochanter has been located (as above), the average 3D position of a suitable set of neighbourhood vertices is calculated and replaces

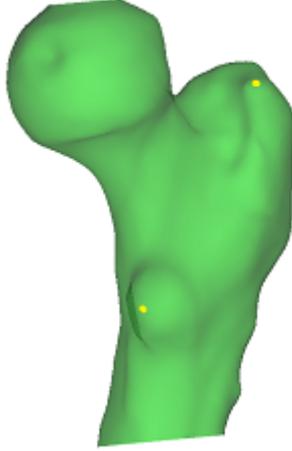


Figure 8: Example of selected points.

the distinguished point. The neighbourhood vertices are those vertices that lie within a distance threshold (15 mm) of the chosen vertex and have Gaussian curvature above a threshold ( $0.001 \text{ mm}^{-2}$ ).

The weights  $\lambda_{K,LT}$ ,  $\lambda_{H,LT}$ ,  $\lambda_{K,GT}$ ,  $\lambda_{H,GT}$ , and  $\lambda_{\mathbf{n}}$  must be carefully chosen, as the performance of the algorithm is very sensitive to their values. The set of values shown in Table 1 was found to give good results and is used for all further experiments.

Parameter	Value
$\lambda_{K,LT}$	10000
$\lambda_{H,LT}$	1000
$\lambda_{K,GT}$	0
$\lambda_{H,GT}$	20
$\lambda_{\mathbf{n}}$	10

Table 1: Values of weighting factors used in the distinguished point detection scheme.

Once these points have been selected, it is necessary to incorporate them into the existing ICP cost function. This is achieved by making sure that corresponding distinguished points on the two surfaces are always paired with each other (instead of using the true closest point), and then giving increased weight,  $\lambda_{DP}$ , to distance between these pairs when evaluating the cost function. Weights in the range  $40 \leq \lambda_{DP} \leq 100$  were found to give reasonably good results. If the weight is lower than this, the distinguished points do not have much influence on the process and behaviour is similar to that when the standard ICP cost function is used. Above this range, the process tends to overfit to the exact location of the distinguished points at the expense of the alignments of other points on the surface.

### 3 Evaluating Registration Quality

Another shortcoming of the existing registration process is that in some situations it causes distortion of the mesh that seems intuitively unnecessary to produce a good registration. This can be seen clearly in Figure 9.

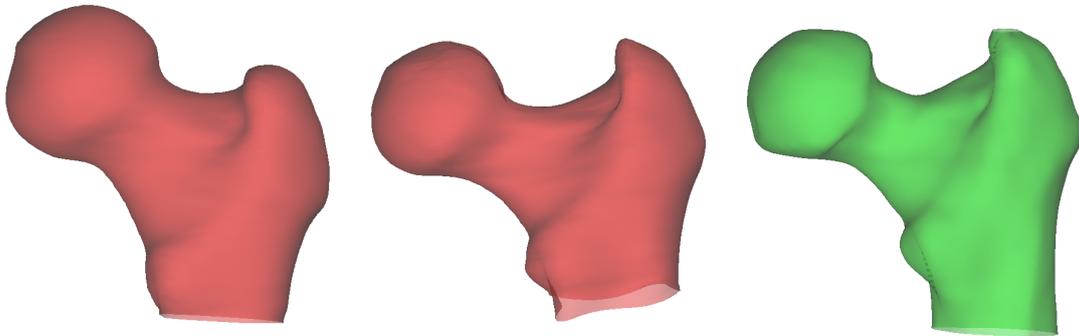


Figure 9: Example of a registration with unneeded deformation: the canonical femur before registration (*left*), the registered canonical femur, showing warping of the shaft (*centre*), and the target femur (*right*).

The two surfaces in this figure are aligned very well in terms of registration error (distance between closest points), and in terms of the alignment of the salient anatomical features. However, the non-uniform deformation of the shaft in the ‘vertical’ (*proximal-distal*) direction (easily seen by looking at the shaft rim in the figure) is intuitively not necessary to align the two surfaces. Because there is little curvature of the shaft in this direction, the transformation could deform the shaft significantly in this direction with only small changes in registration error and alignment of distinguished points. However, we wish for our registration process to give us a solution without such distortion.

There are a number of possible factors contributing to the failure of the existing process:

- *Overfitting* to the target surface.
- The fact that the B-spline technique is not sufficiently localised.
- The inability of the current technique to model *gross* shape changes, such as the angle of the neck.
- The method used to select the closest points in the matching stage.

The causes of warping and ways to reduce it are discussed further in Sections 4 and 5. The remainder of this section will be concerned with developing a technique for *quantifying* this unwanted distortion, with a view to then evaluating methods to reduce it. In §3.2 a method based on the coplanarity of points on contours of the surface is presented.

### 3.1 Previous Work

There is remarkably little existing literature concerned with quantifying the distortion undergone by meshes during transformations or with evaluating registration processes based on this idea. Some refer to methods for specific transformations using functions of the transformation parameters. For example, Szeliski and Lavallée [25] use functions of the positions of spline control points as a measure of distortion to regularise their optimisation algorithm. The same authors also suggest a general form using the difference of parameter vectors. If the parameters of a transformation may be placed into a vector  $\mathbf{v}$ , and the parameters of the identity of the transformation are  $\mathbf{v}_0$ , then a measure of distortion is:

$$D(\mathbf{v}) = \|\mathbf{v} - \mathbf{v}_0\|^2$$

However, this would not provide meaningful comparisons between different transformations parametrised in different ways. We also wish to try and avoid penalising ‘useful’ deformation needed to correctly register the two surfaces.

To the best of our knowledge no such metric has been previously described.

### 3.2 Mesh Distortion Metric

Here we present an original mesh distortion metric, based on measuring how close sets of points are to being coplanar. The rationale behind this is that the transformation should distort the surface such that points on a planar contour around the surface should also lie roughly in a plane after the transformation. This is particularly important around the shaft as in most cases this deformation is unnecessary, such as observed in the example in Figure 9.

Given a description of a plane, the first step is to find a closed contour of points around the surface lying close to where this plane intersects the surface. This is done by finding an initial *seed* point that lies closest to the plane, and then moving around the surface clockwise (when viewed along the direction of the normal of the plane), at each step finding the closest point to the plane that lies clockwise from the current point. This is shown in Algorithm 1. Additionally, there are mechanisms to cope with contours that cross a rim in the surface, but these have been omitted from Algorithm 1 for clarity.

After the contours have been found on the undistorted surface, the average distance of the points on the contour from the plane is found. Then the *Levenberg-Marquardt* algorithm is used to fit a new plane to the same points on the *transformed surface*, and the average distance of the transformed points from this plane can be found in the same way. The comparison of these values gives a measure of the distortion of the mesh.

A grid of nine such contours is defined on the canonical surface, with three perpendicular to each of the *natural axes*, see Figure 10. This can be used to give a measure of the

**Input:**  $i_0 \in \mathbb{N}_0$  a seed vertex,  $\mathbf{a} \in \mathbb{R}^3$  the plane normal,  $b \in \mathbb{R}$  the plane offset  
**Output:** a list of vertices on the contour

```

 $i \leftarrow i_0$  initialise at the seed point
repeat
   $\mathbf{d} \leftarrow \mathbf{a} \times \mathbf{n}_i$  define the search direction
  for all vertices  $ii$  in the neighbourhood of  $i$  do
    if  $((\mathbf{p}_{ii} - \mathbf{p}_i) \cdot \mathbf{d} > 0)$  consider only vertices in this direction then
       $\epsilon \leftarrow |(\mathbf{p}_{ii} \cdot \mathbf{a}) - b|$  distance of point  $ii$  from plane
      if  $(\epsilon < \epsilon_{min})$  then
         $\epsilon_{min} \leftarrow \epsilon$ 
         $j \leftarrow ii$  make this the next point
      end if
    end if
  end for
   $i \leftarrow j$  advance to next point
until  $i = i_0$  we return to the seed point

```

Algorithm 1: Algorithm for finding a contour.

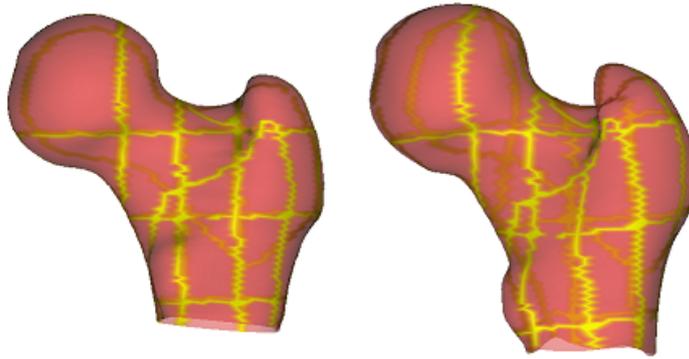


Figure 10: Contours on the canonical surface: before registration (*left*) and after registration (*right*).

overall warping of the mesh, or the contours can be examined individually (in particular the contour located on the lower shaft). This method is used to evaluate a number of methods introduced in Sections 4 and 5.

### 3.3 Synthetic Data and Ground Truth Error

The distortion undergone by a mesh during registration is just one aspect of the quality of a registration process. Another important consideration is the extent to which the matched points correspond anatomically. In general, this is also very difficult to measure meaningfully. However in the case where the *ground truth* correspondence between the two surfaces is known, it can be evaluated straightforwardly. This makes it possible to use *synthetic data* to evaluate registration processes.

A dataset consisting of 100 synthetic surfaces and a ‘canonical’ synthetic surface was

used to evaluate the registration processes. These surfaces have a very simple shape, mimicking the shaft, neck, and head of a femur. Each of the surfaces has a different shaft-neck angle and a different head length (as these are two of the most important variations between femurs, see §4), but they are otherwise similar. There is a known one-to-one correspondence between points each of the surfaces. Two such surfaces are shown in Figure 11.

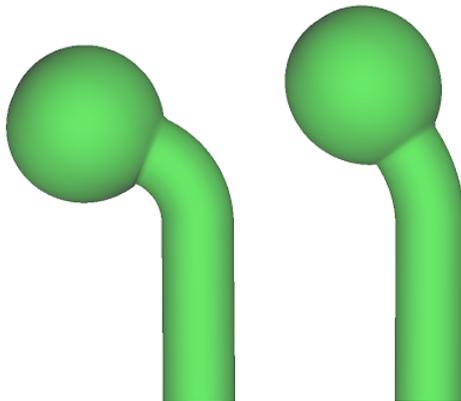


Figure 11: Example surfaces from the synthetic dataset.

The *ground truth error* between two registered synthetic surfaces can be found by finding the average distance between the true correspondence pairs after registration (rather than the distance between closest points). This gives another criterion against which to assess the performance of different registration techniques. Unfortunately however, due to the simplified nature of the surfaces, it is not appropriate to use the distinguished point method with the synthetic dataset.

## 4 Alternative Transformations

In this section, we consider changes to the existing transformations in order to address the issue of unwanted warping as introduced in §3.

Although affine and B-spline transformations have been shown to work well with the sorts of deformations or variations found in breast and brain tissue [10, 26], they are not designed to cope with the large variations that are seen amongst femurs. In particular, the angle made between the head and the neck of the femur varies significantly between individuals, as does the length of the neck (see Figure 12 for an example). In this situation, the ‘ideal’ transformation would change the angle of the neck in an ‘almost rigid’ fashion and stretch it along its axis whilst affecting the rest of the femur (shaft etc.) as little as possible. B-spline deformations, especially those with few control points, are not well suited to carrying out such ‘almost rigid’ deformations of parts of surfaces because they warp the space in which the surface is embedded.

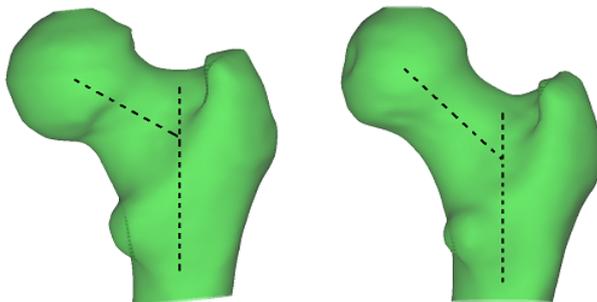


Figure 12: Two femurs with very dissimilar neck lengths and head-neck angles.

The B-spline stage used in the existing process uses a  $4 \times 4 \times 4$  grid of control points, which is not sufficiently *localised* to prevent, for example, the registration of the head and neck interfering with that of the shaft. In fact, in a volumetric B-spline deformation, the transformed position of a point is affected by the positions of the surrounding 64 control points. Therefore, in our grid, the warped position of each vertex on the surface is affected to some extent by the positions of *all* the control points. Consequently, when control points move to make large scale changes (such as aligning the head and neck), other areas of the surface (such as the shaft) will also be affected, particularly if the cost function is fairly indifferent to this movement.

This could be easily improved by increasing the number of control points in the grid. However, adding one more plane of control points in each dimension to give a  $5 \times 5 \times 5$  grid would lead to an optimisation problem with 375 degrees of freedom for each ICP iteration. It is likely that several more control points would be needed to get the desired degree of localisation. The computations quickly become impractical, especially when the process is required to carry out cohort studies on a large number of femurs. Such an optimisation problem is also likely to encounter serious problems with local minima.

The remainder of this section introduces constraints to the B-spline transformation in an attempt to reduce the warping, and discusses other types of transformation that may be able to reduce this problem.

## 4.1 Previous Work

One way to prevent warping is to simply constrain or regularise the deformation stage. Simple regularisation schemes for B-splines are offered by Rueckert et al. [10, 26]. However, these also reduce the ability of the transformations to correctly register the surfaces. A method to constrain the current transformations is discussed in §4.2.

Many authors have considered various polynomial or spline-based transformations, such as Szeliski and Lavallée’s Octree splines [25], Bookstein’s thin-plate splines [27] or simple low-order polynomial transformations [7, 17]. However such approaches are likely to come up against similar shortcomings to the existing B-spline process.

Xie and Farin present a *hierarchical B-spline* scheme that allows for more localised deformation where it is needed without incurring the high computational cost of using a very high number of control points [28]. This works by starting with a  $4 \times 4 \times 4$  grid exactly as in the current scheme, then subdivides this to form new B-spline grids with smaller spacings to match areas where the registration error is still high. However, since it is often the initial large deformations which are contributing to the warping in the existing process, it is unlikely that this approach would produce significant improvements for our purposes.

A more promising approach is found in the *locally affine* method of Feldmar and Ayache [29]. This method abandons the standard free-form deformation approach of warping the space in which a surface is embedded, and instead finds deformations to apply to each vertex of the surface on an individual basis. This involves an optimisation over the 6 DoF space of rigid-body transformations for every vertex on the canonical surface, but this is far faster than conducting a single optimisation in a very high dimensional space (as would be required by a B-spline with many control points). Smoothness of the transformation is maintained by averaging the deformation over local patches of the surface. The key advantage of the algorithm is that the degree to which the deformation is localised is controlled by the extent of this smoothing. The locally affine method was implemented and tested as described in §4.3.

## 4.2 Constrained (‘Physical’) Transformations

An existing initial attempt to reduce the problem of mesh distortion, which is evaluated in this project, was based on constraining the existing global affine/local B-spline transformation. By simply reducing the number of degrees of freedom a transformation has, the amount of distortion that it can create in a mesh is reduced. However, this must

be balanced against the ability of the transformation to register the two surfaces. The resulting transformation is referred to as the ‘*physical*’ transformation, and consists of two parts: a global linear transformation followed by a constrained B-spline deformation stage.

#### 4.2.1 Global ‘Physical’ Transformation

The existing process uses a 12 DoF affine transformation as the global stage of the registration. These 12 degrees of freedom can be interpreted as three *translational* components, three *rotational* components, three *scaling* components and three *shearing* components. The transformation can be represented by a *general* ( $3 \times 3$ ) matrix  $\mathbf{A}$  and a translation vector  $\mathbf{b}$ :

$$\mathcal{T}(\mathbf{p}) = \mathbf{A}\mathbf{p} + \mathbf{b}$$

Although planar points remain planar under an affine transformation, the shearing components can still introduce unwanted deformation into the mesh. The ‘physical’ transformation constrains the affine transformation by eliminating the three shearing components, giving a 6 DoF linear transformation plus a translation (and therefore a total of 9 DoF). Equivalently, this transformation can be interpreted as a generalisation of a rigid-body transformation to include three *anisotropic scaling* components. This can be represented by a ( $3 \times 3$ ) matrix  $\mathbf{M}$  and a translation vector  $\mathbf{b}$  applied to a point  $\mathbf{p}$ :

$$\mathcal{T}(\mathbf{p}) = \mathbf{M}\mathbf{p} + \mathbf{b}$$

where  $\mathbf{M}$  is constrained to have a specific form, determined by the three *Euler angles*,  $\alpha$ ,  $\beta$ ,  $\gamma$  and the three *scaling factors*  $\sigma_1$ ,  $\sigma_2$  and  $\sigma_3$ , which together with the translational components are the nine parameters of the transformation. Taking  $s_\alpha$  and  $c_\alpha$  to be shorthand for  $\sin(\alpha)$  and  $\cos(\alpha)$  respectively, the form of  $\mathbf{M}$  is given by:

$$\mathbf{M} = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$$

#### 4.2.2 ‘Physical’ B-spline Transformation

The existing local B-spline transformation uses a  $4 \times 4 \times 4$  grid of control points, with each control point having 3 DoF representing its position in 3D space. To constrain this deformation and prevent it from causing warping of the shaft, the lower two of the four horizontal planes of the control point grid (which have a large influence on the position of vertices on the shaft) are constrained to remain planar (but the planes themselves may move relative to each other, whilst maintaining the same orientation) as shown in Figure 13. This reduces the total number of degrees of freedom from 192 to 162 and

reduces the ability of the transformation to warp the shaft, whilst leaving much of the flexibility around the head and neck needed to register the surfaces.

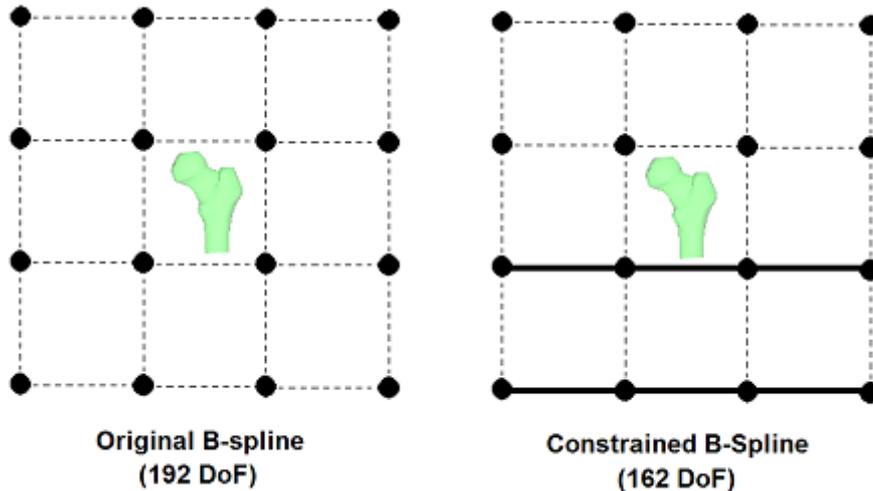


Figure 13: In the ‘physical’ B-spline deformation, the control points in the lower two planes are constrained to stay within their planes (though the planes themselves may move with a single degree of freedom).

### 4.3 Locally Affine Deformation

The locally affine method introduced by Feldmar and Ayache [29] is *non-parametric* in that the transformation cannot be represented by any single global set of parameters. Instead the position of every point on the surface is independent and is stored separately.

The algorithm works by considering small areas of the surface at a time. It first finds the rigid body (6 DoF) transformation that maps each vertex along with a small set of surrounding vertices to their closest points on the other surface. Then these rigid-body transformations are smoothed over the surface to ensure that the overall transformation is smooth. This gives a set of (generally different) affine transformations at each vertex on the surface, which gives rise to the name *locally affine*. Each vertex is then deformed by its own affine transformation, and the process repeats in an iterative manner using the ICP framework.

There is one free parameter of the process,  $R$ , which controls the radius of the group of points considered when finding the rigid-body transformations. The larger the value of  $R$ , the smoother the final surface will be and the less able the deformation will be to match the two surfaces. However, if  $R$  is too small there is little smoothness constraint and deformations will overfit one surface to the other.

In detail, the algorithm consists of the following stages:

1. For each vertex  $i$  on surface  $\mathcal{S}_A$ , find the set of points on surface  $\mathcal{S}_A$  that lie within a distance  $R$  of  $i$  (including the vertex  $i$  itself). This is the *sphere set* of  $i$ , denoted

$\mathcal{V}_i$ . To speed up the algorithm, it is assumed that this set will not change during the deformation process, which is reasonable because the smoothing ensures that vertices will not move significantly relative to their neighbours.

2. For each vertex  $i$  on surface  $\mathcal{S}_A$ , find the closest point on the surface  $\mathcal{S}_B$  as in the usual ICP method (or one of the matching methods discussed in §5).
3. For each vertex  $i$ , find the rigid body transformation (described by the rotation matrix  $\mathbf{R}_i^{(t)}$  and the translation vector  $\mathbf{t}_i^{(t)}$ ) that, if applied to the whole surface, would minimise the distance (in the least squares sense) between the vertices in  $\mathcal{V}_i$  and their matched vertices on surface  $\mathcal{S}_B$ . Here,  $t$  represents the time index within the iterative process. The optimisation is performed using the *Levenberg-Marquardt* algorithm to find the six parameters of the rigid-body transformation.
4. Smooth the rigid body transformations  $\{\mathbf{R}_i^{(t)}, \mathbf{t}_i^{(t)}\}$  to give the locally affine transformations at each point  $\{\mathbf{A}_i^{(t)}, \mathbf{b}_i^{(t)}\}$ , where  $\mathbf{A}_i^{(t)}$  is a general  $(3 \times 3)$  matrix and  $\mathbf{b}_i^{(t)}$  is a translation vector. This is done using a weighted average over the transformations at all the vertices in a vertex's *sphere set* as follows:

$$\mathbf{A}_i^{(t)} = \sum_{j \in \mathcal{V}_i} \lambda_j^{(t)} \mathbf{R}_j^{(t)}$$

$$\mathbf{b}_i^{(t)} = \sum_{j \in \mathcal{V}_i} \lambda_j^{(t)} \mathbf{t}_j^{(t)}$$

where the weighting factor,  $\lambda_j$  accounts for the distance of vertex  $j$  from  $i$ ,

$$\lambda_j^{(t)} = \frac{\left(1 - \frac{\|\mathbf{p}_j^{(t)} - \mathbf{p}_i^{(t)}\|}{C}\right)}{\sum_{k \in \mathcal{V}_i} \left(1 - \frac{\|\mathbf{p}_k^{(t)} - \mathbf{p}_i^{(t)}\|}{C}\right)}$$

$C$  is a constant describing the characteristic scale of the surface, taken here to be the maximum distance between the estimated centre of mass and the surface.

5. Apply the locally affine transformations to the points, so that at the start of the next iteration ( $t + 1$ ):

$$\mathbf{p}_i^{(t+1)} = \mathbf{A}_i^{(t)} \mathbf{p}_i^{(t)} + \mathbf{b}_i^{(t)}$$

6. Return to step 2, or terminate.

Since this method gives an inherently localised transformation, it is used after an initial global transformation (such as an affine transformation or the ‘physical’ transformation of §4.2) and replaces the local B-spline stage of the original registration process.

## 5 Alternative Methods for Finding Matched Points

An alternative way to alter the *similarity criterion* of an ICP-based process is to change criterion that is used to find point pairs during the matching stage. In standard ICP, matched points are simply chosen to be the closest point on the other surface in 3D space. When using other methods, the matched points will not generally be the closest points in 3D Euclidean space. Therefore from now on, we will refer to finding *matching points* rather than *closest points* to avoid ambiguity. The point matches are treated exactly the same during the transforming stage regardless of how they are found during the matching stage.

The definition of the problem is, given a transformed vertex  $i$  with position  $\mathcal{T}(\mathbf{p}_i)$  on the canonical surface,  $\mathcal{S}_A$ , find a matching vertex  $i^*$  with position  $\mathbf{q}_{i^*}$  on the target surface,  $\mathcal{S}_B$ .

There are two reasons for exploring these alternative methods. Firstly, they provide an alternative to the use of distinguished points for ensuring that anatomically corresponding points are brought into correspondence, but they have the potential advantage of ensuring similarity between *all* points on the surfaces.

Secondly, using these methods may help prevent unnecessary mesh distortion caused by the standard closest point method (see §3).

The current method for selecting closest points uses a *spatial hashing* of points to make the process of searching the points far quicker. In this section, three different schemes are described to improve the point matching whilst still using the spatial hashing to give efficient computation.

### 5.1 Previous Work

Authors have used a variety of methods to match points, usually based on matching differential geometric properties of points as well as their physical proximity. Feldmar and Ayache [29] match by finding the closest point in an 8-dimensional space composed of the 3 components of the vertex’s position, the 3 components of its normal, and its two principal curvature values (although one could also imagine using Gaussian and mean curvature to similar effect). This means that matched points have similar normal and curvature properties as well as positions, with clear advantages for ensuring that matches are anatomically meaningful. However, searching efficiently in an 8-dimensional space is a difficult problem. Münch et al. [30] simplify this by finding the point that minimises a weighted combination of the Euclidean distance between two points and a term quantifying the difference of their surface normals. We will refer to this as the *normal-weighted* method.

Rusinkiewicz and Levoy [31] review a number matching techniques, but some are only suitable for range images and not surfaces. These are mostly variants on *projecting*

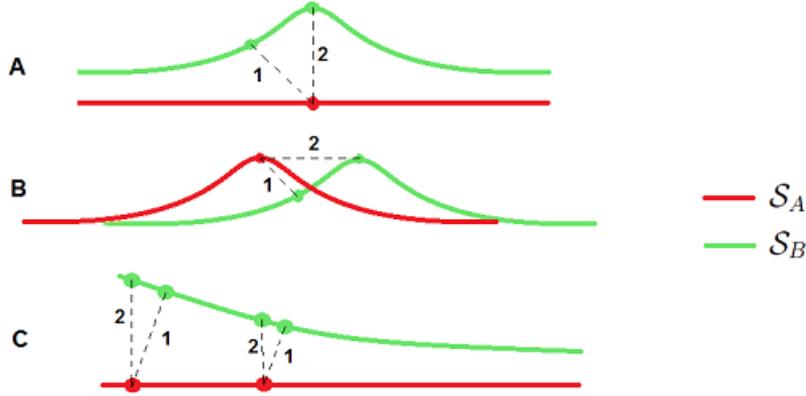


Figure 14: Illustration of the advantages of using alternative matching schemes.

points from one surface to the other along the normal of one surface, a technique used by a number of other authors such as Chen and Medioni [9]. We will refer to this method as *normal-shooting*.

The potential advantages of using these techniques are illustrated in Figure 14. In Figure 14A, the existing closest point process would choose the point pair labelled 1, which could lead to unnecessary warping of  $\mathcal{S}_A$  to the left or right. However using either the *normal-weighted* or *normal-shooting* method would lead to the pair labelled 2, a more intuitive match that would prevent such warping.

In Figure 14B, using closest points would lead to the point pair labelled 1. This could lead to a situation such as that in Figure 3 where the process fails to bring anatomically corresponding points together. Using a *normal-weighted* matching scheme would lead to the point pair labelled 2 and overcome this problem. The *normal-shooting* scheme would not help in this case.

Finally, Figure 14C shows a case where the *normal-shooting* method is useful. Using the closest point matching scheme would lead to the pairs labelled 1. The result may be that  $\mathcal{S}_A$  is deformed to the right as it is being registered to  $\mathcal{S}_B$ . This is unwanted deformation. Using the normal shooting method would lead to the pairs labelled 2 and could therefore prevent such warping.

Rusinkiewicz and Levoy also suggest ignoring point matches where the matched point lies on a rim of the target surface. This very simple technique is important when the canonical surface extends beyond the target surface in some places (as is the case with many surfaces in the dataset) as it prevents the canonical surface from ‘bunching up’ where there is no corresponding part of the target surface. This is illustrated in Figure 15, where the closest point scheme would lead to the matches shown. The result would be that, depending on how localised the transformation is, the red surface would be shifted to the right, which in many cases is not wanted. This can be avoided by discarding the two leftmost matches.

Within the field of Computer Graphics, matching surfaces that undergo *articulated*

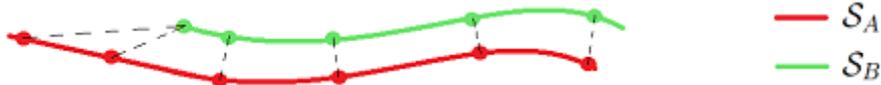


Figure 15: Illustration of the advantages of ignoring rim points.

motion (e.g. bending of the human arm) is a common problem, and this has given rise to a different breed of algorithms that find matches between points on two surfaces using a variety of sophisticated methods, sometimes without finding an underlying transformation. The key to solving such problems is considering *geodesic* distances (i.e. distance along the surface of a mesh) rather than *Euclidean* distances. This approach can cope very well with large deformations of the mesh if they are close to being *isometric* in terms of geodesic distance. This is the case with articulated deformations, such as motion of the arm relative to the body in a model of a human body, where Euclidean distances between two points can be very different depending on the pose of the body, but geodesic distances remain approximately the same. This property is useful to us because it potentially provides a way to account for the ‘almost rigid’ deformation of the neck relative to the shaft.

Huang et al. describe one such algorithm [32]. Their approach is based on the ICP method, but they match points based on similarity of a feature vector consisting of 3D position and principal curvatures and then *prune* this set of matches to produce a set of that are consistent in terms of geodesic distance.

The *correlated correspondence algorithm* of Anguelov et al. [33] is another example. This algorithm defines a joint probability distribution over the entire set of matches using the deformation relative to neighbouring points and the consistency of the geodesic distance between pairs. In this way, the correspondences of neighbouring points become correlated. They then use numerical inference to produce a consistent set of point matches.

Some authors have chosen to estimate the correspondence between two surfaces via an intermediate representation in the complex plane [34, 35, 36]. Firstly, a *conformal mapping* is estimated from each surface to a *canonical* representation in the complex plane (a process known as *conformal flattening*), and then a transformation between the canonical representations of the two surfaces is estimated. The advantage of this approach is that it is *geodesic* distances rather than *Euclidean* distances between points that govern the mapping of the surface to the canonical domain.

Gu and Vemuri [34] find the transformation in the canonical domain that minimises the difference between the *conformal factor* (change in area when undergoing a conformal mapping) and the mean curvature at each point on the surface. Lipman and Funkhouser [35] and Boyer et al. [36] use a very similar procedure, but vote for matches by sampling random transformations within the canonical domain. This approach is of

particular interest to us as Boyer et al. have shown it to provide useful mappings between the surfaces of bones such as teeth and radii. Their scheme is discussed further in §5.6.

It would be very interesting to evaluate the application of the other techniques to our problem, but it was unfortunately beyond the scope of this project to implement and test them.

## 5.2 Normal Transformations

In order to incorporate surface normal information into the point matching process, it is necessary to be able to find the normals of the *transformed* canonical surface. When a surface undergoes a transformation  $\mathcal{T}$ , the normals undergo a transformation which will be denoted  $\mathcal{N}_{\mathcal{T}}$ .

When  $\mathcal{T}$  is a linear transformation, there is a convenient form for the normal transformation, given by Turkowski [37]. If the transformation  $\mathcal{T}$  is represented by a  $(3 \times 3)$  matrix  $\mathbf{A}$  (the surface normals are invariant to the translation part of the transformation so this may be ignored), then the normal transformation is represented by its *inverse transpose*,  $\mathbf{A}^{-T}$ . This does not in general result in a unitary (length-preserving) transformation, so to get unit normals it is necessary to re-normalise after the transformation. Therefore the transformed normal at time index  $t$  is given by

$$\mathbf{n}_i^{(t)} = \mathcal{N}_{\mathcal{T}}^{(t)}(\mathbf{n}_i^{(0)}) = \frac{(\mathbf{A}^{(t)})^{-T} \mathbf{n}_i^{(0)}}{\|(\mathbf{A}^{(t)})^{-T} \mathbf{n}_i^{(0)}\|}$$

For many types of transformation however, it is impractical to find and use an expression for  $\mathcal{N}_{\mathcal{T}}$ , so the normals are simply re-estimated after the transformation by considering normals of the incident triangles on the transformed surface.

## 5.3 Normal Weighted Point Matching

This scheme is inspired by that of Münch et al. [30], but it is formulated in a way that allows the spatial hashing to be used to improve efficiency. Only first-order information (i.e. surface normals) are considered for simplicity, and because the advantage of using second-order information (i.e. curvatures) is not clear in this case. However, the scheme could be easily extended to include curvature information. For each vertex  $i$  on  $\mathcal{S}_A$ , the matching vertex  $i^*$  on  $\mathcal{S}_B$  is chosen such that:

$$i^* = \arg \min_j \left\{ \|\boldsymbol{\delta}_{ji}\| + \lambda_n \left( 1 - (\mathbf{r}_j \cdot \mathbf{n}_i^{(t)}) \right) \right\}$$

where  $\boldsymbol{\delta}_{ji}$  is the 3D Euclidean distance between the two points,

$$\boldsymbol{\delta}_{ji} = \mathbf{q}_j - \mathbf{p}_i^{(t)}$$

The first term in this cost function represents the distance between the two vertices' positions (as in the standard ICP process), and the second term represents the dissimilarity between their normal vectors, weighted by a non-negative weighting factor  $\lambda_{\mathbf{n}}$ .

The spatial hashing scheme only hashes points within a limited distance of the point under consideration. This limits the physical distance between matched points even when the weighting is heavily towards the normal direction (large  $\lambda_{\mathbf{n}}$ ). This is reasonable since the initial alignment step should have brought the surfaces comfortably within this distance of each other already.

A range of different behaviours can be achieved by altering  $\lambda_{\mathbf{n}}$ . Weight values in the range  $0 < \lambda_{\mathbf{n}} \leq 100$  capture the range of behaviours between standard closest point matches, and matches strongly in favour of normal alignment. Increasing the weight indefinitely has no effect because of the finite spatial hashing grid size.

## 5.4 Normal 'Shooting' Point Matching

This scheme is inspired by Chen and Medioni's [9], but again it is formulated in a way that allows it to take advantage of the existing spatial hashing scheme. Instead of simply projecting the point from  $\mathcal{S}_A$  to  $\mathcal{S}_B$  along the normal of  $\mathcal{S}_A$ , a weighted cost function is used which penalises distance between the two points in the plane perpendicular to the normal of  $\mathcal{S}_A$ , as follows:

$$i^* = \arg \min_j \left\{ \|\delta_{ji}\| + \lambda_{\mathbf{n}} \|\delta_{ji} - (\delta_{ji} \cdot \mathbf{n}_i^{(t)}) \mathbf{n}_i^{(t)}\| \right\}$$

Here, the second term of the cost function represents the component of the difference between the two points,  $\delta_{ji}$ , that is perpendicular to the normal of surface  $\mathcal{S}_A$ ,  $\mathbf{n}_i^{(t)}$ . The range  $0 < \lambda_{\mathbf{n}} \leq 5$  captures the range of behaviours between closest point matches, and matches strongly in favour of points along the normal direction.

Another advantage of assigning some weight to the 3D Euclidean distance (rather than using 'pure' normal shooting) is that the problem of finding points on the 'other side' of the surface is reduced.

## 5.5 Ignoring Rim Point Matches

This is a very straightforward improvement to the existing point matching strategy, described by Rusinkiewicz and Levoy [31]. Once the matched points have been found (using any method), the matches in which the point on  $\mathcal{S}_B$  lies on the rim of mesh  $\mathcal{S}_B$  are discarded. Note that this can be used in combination with either of the above two methods, although it is less likely to be effective in these cases because the points on the canonical surface that are matched to points on the rim of the target surface are no longer necessarily the points that lie beyond the rim of the mesh of the target surface.

Using this method does however create problems when mapping thickness data onto the canonical femur. Even if rim matches are not ignored by the mapping process, points on the canonical surface that do not overlap with the target surface will share matched points on the rim of the target surface. The effect will be extension of the rim thickness value across the non-overlapping part of the canonical surface. The context of the experiments must therefore be considered when deciding whether to use this option.

## 5.6 Conformal Flattening Method

Here we describe the method of Boyer et al. [36], which is underpinned by the closely related work of the co-authors Lipman and Funkhouser [35]. The method is involved and mathematically complex, and shall not be treated in detail here (an excellent description is given by Lipman and Funkhouser in their paper). In essence, the algorithm works by mapping the two surfaces to a *canonical representation* in the complex plane (not to be confused with our unrelated *canonical model* of the femur). Then a search is conducted of transformations between the two canonical representations, rather than between the two surfaces themselves. The transformations considered are those in the *Möbius Group*, which is the set of one-to-one transformations that map the complex plane to itself isometrically. Möbius transformations have the simple form:

$$m(z) = \frac{az + b}{cz + d}$$

where  $a, b, c, d \in \mathbb{C}$  are complex numbers such that  $ad - bc \neq 0$ . A unique Möbius transformation that interpolates any three given distinct points may be computed in closed-form. It is this result that underpins the algorithm.

The steps involved are:

1. A *discrete harmonic function* and a *conjugate harmonic function* are defined on each mesh by solving a sparse system of linear equations. This gives a parametrisation of the surface.
2. For each mesh, this parametrisation is used to create a mapping from each vertex on the surface to a point on the complex plane, a process known as *conformal flattening*. The result is illustrated in Figure 16.
3. In an iterative process, three random points are taken from each of the two flattened meshes, and the Möbius transformation that interpolates the two triplets is computed. For each of these transformations, a correspondence map is produced from mutually closest points. The ‘deformation error’ of this transformation is found by summing the distances between matched points under the transformation, and this

is used to build up a *fuzzy correspondence matrix* between each pair of points by ‘voting’ for correspondences.

4. The *fuzzy correspondence matrix* is processed to give a consistent correspondence map between the two surfaces.

The Boyer et al. MATLAB implementation of this algorithm was used to evaluate it for our purposes<sup>1</sup>. The output of this software is a correspondence map between the two input surfaces (i.e. for each point on  $\mathcal{S}_A$ , the index of the corresponding point on  $\mathcal{S}_B$  is given). Using this correspondence map, the algorithm can be evaluated using the measure described in §3.2.

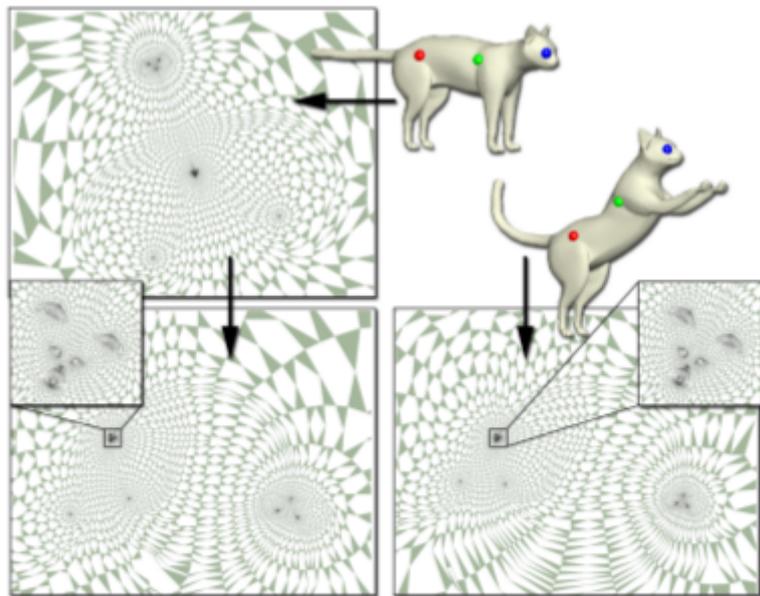


Figure 16: The conformal flattening procedure applied to two cat surfaces (*top left* and *bottom right*), and the two representations aligned using a Möbius transformation (*bottom left*). *Source:* Lipman and Funkhouser 2009.

<sup>1</sup>This is available at <http://www.wisdom.weizmann.ac.il/~ylipman/CPsurfcomp/>

## 6 Results

### 6.1 Registration Failures

Tests for registration failures were conducted by running 100 iterations of the global affine registration on each of 604 femurs, using the distinguished points method and the alternative point matching methods. Registration failures were determined by inspection as described in §2.1. The results are shown in Table 2.

Experiment	Weight	Complete Failures	Lesser Trochanter Failures
Existing Process	-	1 (0.2%)	38 (6.3%)
Distinguished Points	80	5 (0.8%)	0 (0.0%)
Ignoring Rim Points	-	1 (0.2%)	32 (5.3%)
Normal-Weighted Matching <sup>†</sup>	100	1 (0.2%)	9 (1.5%)
Normal-Shooting Matching <sup>†</sup>	5	1 (0.2%)	21 (3.5%)

Table 2: Table of results for the distinguished points experiments.

<sup>†</sup>These experiments also ignored rim points.

### 6.2 Mesh Distortion

Results for the distortion of the mesh were obtained by registering the canonical surface to each surface in the database of 604 target femurs and using the mesh distortion metric presented in §3.2 to evaluate the change in coplanarity of points before and after the registration process. The *overall warping* averages the warping value over each of the nine contours and over all femurs except those where the registration was deemed a failure. This is shown in Figure 17. Because the metric uses contours perpendicular to each of the natural axes, total warping in each of these directions can be measured independently by summing the warping in the three contours perpendicular to that direction and averaging this over the successful registrations. This is shown in Figure 18. Also, as the lower femur is of particular interest to us, the distortion of the single contour on the lower femur is shown in Figure 19, again averaged over the successful registrations.

The parameters used in each of the registrations are shown in Table 3.

Transformation	Iterations	$\lambda_{DP}$	Parameters
Affine	100	60	-
B-Spline	10	60	-
LAD	15	0	$R = 8$

Table 3: Table of parameters for the mesh distortion experiments.

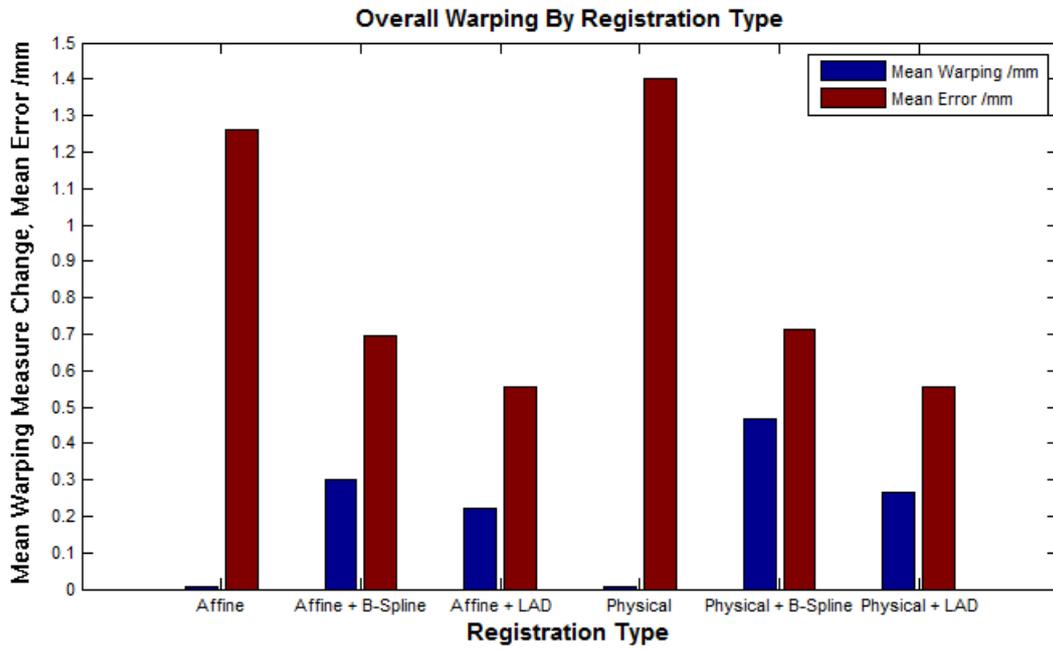


Figure 17: Overall warping and registration error by transformation type.

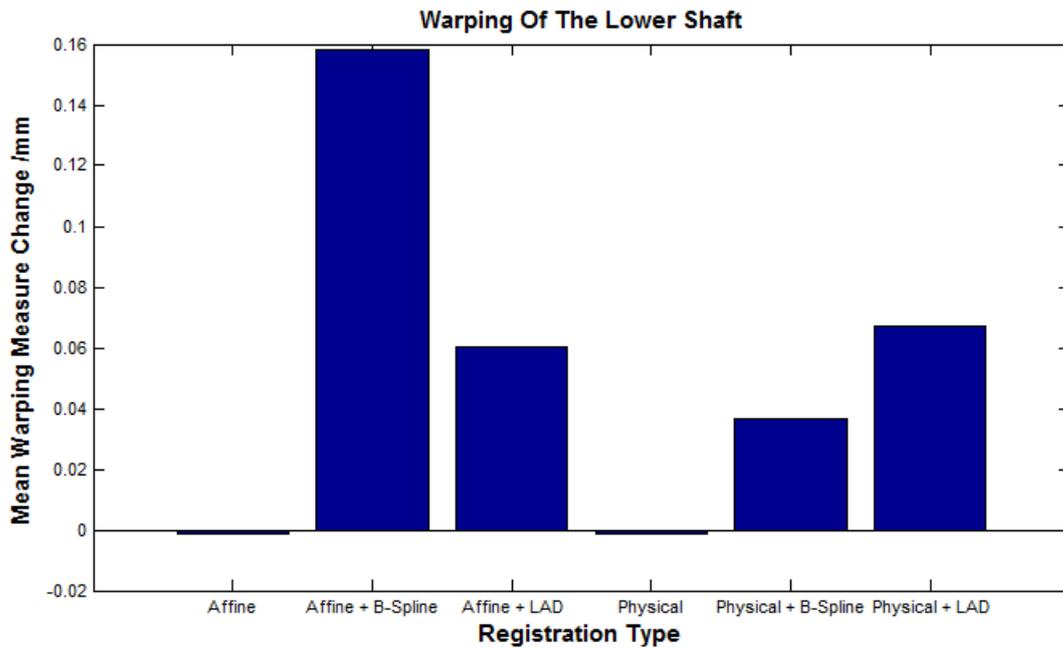


Figure 18: Warping of the lower shaft only by transformation type.

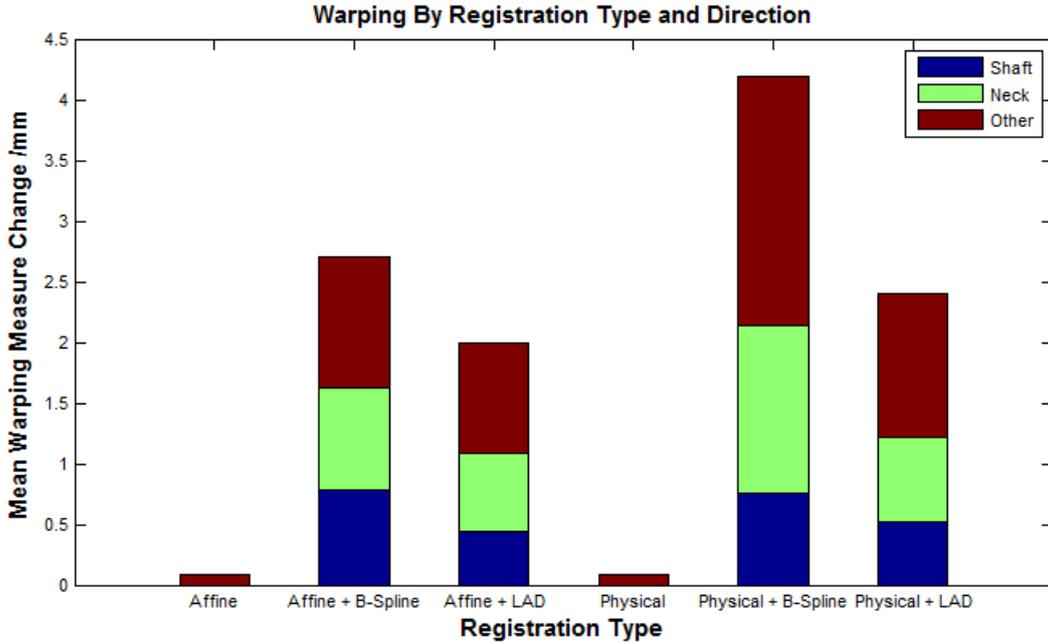


Figure 19: Warping in the *shaft*, *neck* and *other* directions by transformation type.

### 6.3 Alternative Matching Methods

The registration error and mesh distortion were measured after using the *normal-weighted* and *normal-shooting* point matching schemes along with the affine, affine/B-spline and affine/locally affine transformations to register the 604 femurs with the canonical surface. The parameters used were those given in Tables 2 and 3. The results are shown in Figures 20 and 21.

### 6.4 Synthetic Data

The same tests were run on 100 synthetic surfaces using a further synthetic surface as the canonical surface. The parameters given in Tables 2 and 3 were again used. This allowed both the registration error (distance between matched points) and the ground truth error (distance between the ‘true’ corresponding point pairs) to be measured. These results are shown in Figure 22.

The results of an additional test using the *exact* matches are also shown. Here the matched points in the ICP matching stage were forced to be the ‘true’ matches, such that the ground truth error is due only to the inability of the chosen transformation to model the true transformation between the surfaces.

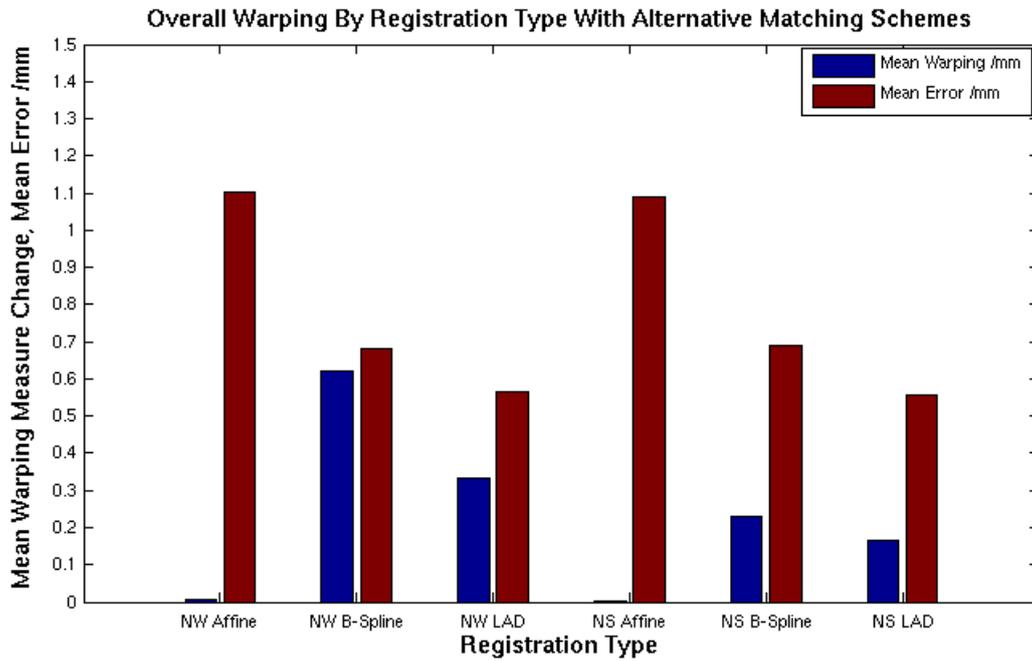


Figure 20: Overall warping and registration error when using alternative matching methods (*NW* normal-weighted, *NS* normal-shooting).

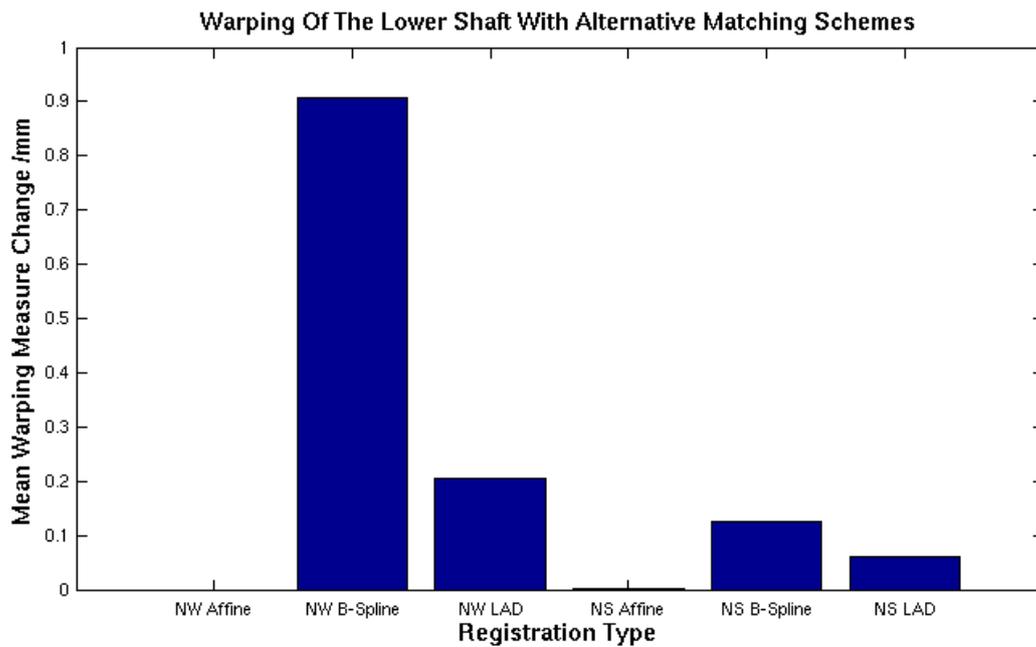


Figure 21: Warping of the lower shaft when using alternative matching methods (*NW* normal-weighted, *NS* normal-shooting).

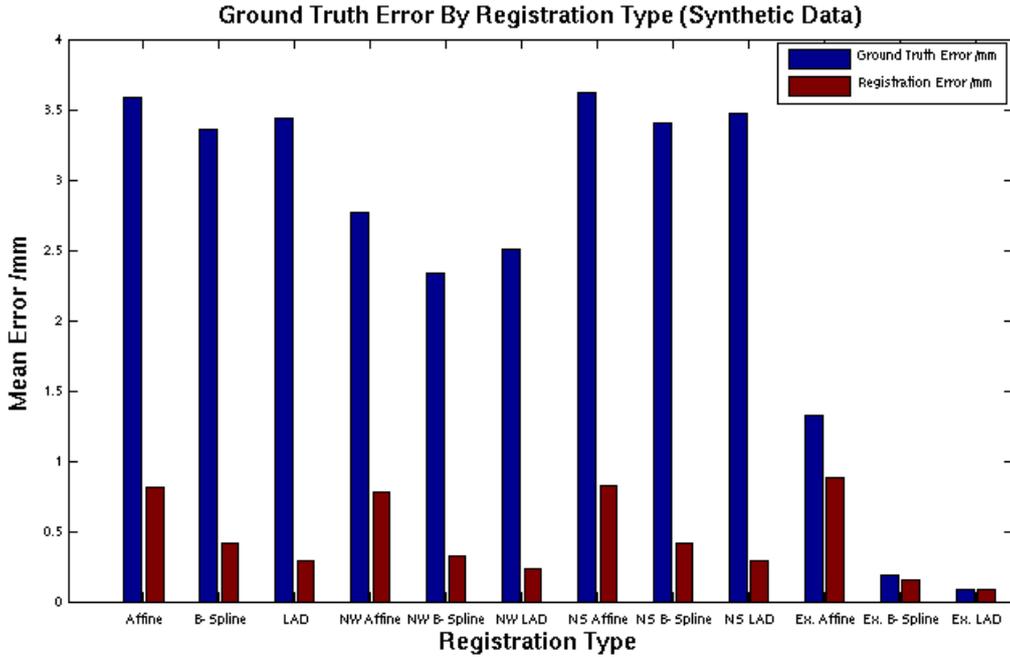


Figure 22: Registration error and ground truth error for the synthetic data experiments (*NW* normal-weighted, *NS* normal-shooting, *Ex.* exact matching).

## 6.5 Conformal Flattening

The conformal flattening method was tested on a small number of surfaces from the dataset in order to get a brief evaluation of its performance. It was found that with some surfaces produced completely nonsensical registrations, whereas others produced intuitively good matches. Since the output of the process is a list of corresponding vertex, the process was evaluated by deforming each vertex on the canonical surface to exactly coincide with its corresponding point on the target surface. Then the warping measurement of §3 can be applied to the canonical surface. Table 4 shows the results for three successful registrations.

Location	Mean Warping Metric /mm
Lower Shaft	2.34
Total Shaft	9.54
Total Neck	10.85
Total Other	11.73
Overall Average	3.57

Table 4: Table of results for the conformal flattening experiments.

## 7 Discussion

### 7.1 Distinguished Points

It can be seen from Table 2 that the distinguished points algorithm of §2.3 successfully removed the problem of lesser trochanter failures. This comes at the expense of introducing a small number of ‘complete failures’ where the distinguished points were detected incorrectly, leading to nonsensical registration. However, the surfaces on which the distinguished point detection failed had very unusual characteristics, as shown in Figure 23.

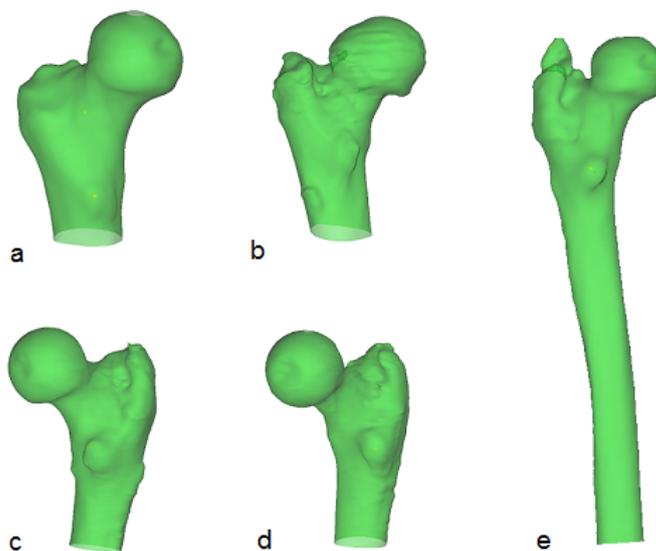


Figure 23: Surfaces for which the distinguished point detection algorithm failed.

In two cases (surfaces **b** and **d** above), failure was due to detection of one of the distinguished points at a small area of very high curvature somewhere else on the surface. These areas are almost certainly the result of an anomaly in the segmentation process rather than a true feature of the scanned femur, and could be removed by smoothing the surfaces in the upstream process. In another two cases (surfaces **c** and **e**), the greater trochanter point was detected at a growth on the greater trochanter. In the final case (surface **a**), detection of the greater trochanter failed due to the very unusual shape of the surface. There is no intuitive greater trochanter point in this case.

Given the low failure rate and the difficult nature of the failure cases, it seems unnecessary to use more sophisticated differential geometry to find distinguished points (for example using better curvature estimates, or using principal directions). One possible improvement would be to train the weighting factors automatically, using a dataset with manually labelled distinguished points.

## 7.2 Alternative Transformations

Figures 17, 18 and 19 show that the physical transformation (§4.2) does indeed reduce the degree of warping of the lower shaft as intended. However, it is also clear that this comes at the cost of a significant increase in the overall warping of the surface. This is because the global physical transformation is less able to match the surfaces in the first stage of the process, so the B-spline stage must warp the surface more to achieve a close match, particularly around the neck and head.

During the experiments it was observed that the global physical transformation stage produced a number of poor registrations, especially on surfaces where the shaft-neck angle or neck length was significantly different from that of the canonical femur (such as the example in Figure 24). Also, the constrained ‘physical’ B-spline may increase warping of the upper parts of the femur by forcing the unconstrained control points to undergo more extreme displacements in order to achieve a match of the head and neck.

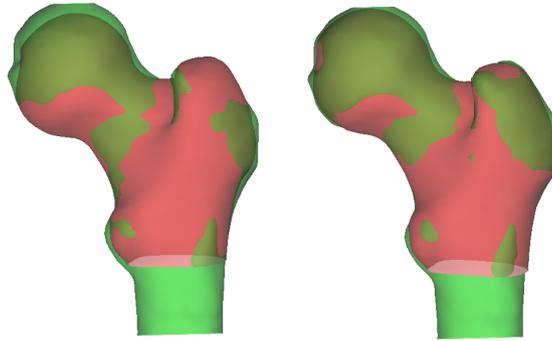


Figure 24: Example of a poor global physical registration (*left*) and the affine registration of the same surfaces (*right*).

The locally affine transformation method (§4.3) gives promising results. Due to its more localised nature, it results in reduced warping over the entire surface when compared to the B-spline transformation. The process was observed to give intuitively sensible matches and the obvious warping of the shaft was not seen to the same extent as it is with B-splines process. Furthermore, unless a relatively large radius parameter is used (around  $R > 15$ ), the registration process using the locally affine transformation is often slightly quicker than using B-splines. The locally affine and B-spline methods performed similarly in terms of ground truth error in the synthetic data experiments, although the B-spline method generally gave slightly better performance.

However, the locally affine method used in the experiments does not really address the fundamental problem of modelling the changes in neck length and shaft-neck angle. Further work could consider using a multi-stage locally affine process starting with a large radius parameter, in which case deformations approximating the rigid rotations may be possible, and then moving to a smaller radius stage to give localised matching.

### 7.3 Alternative Matching Schemes

Ignoring the rim matches as described in §5.5 solves six of the original registration failures from the dataset. As expected these are cases where the canonical surface extends below the target femur, such as the example shown in Figure 25.

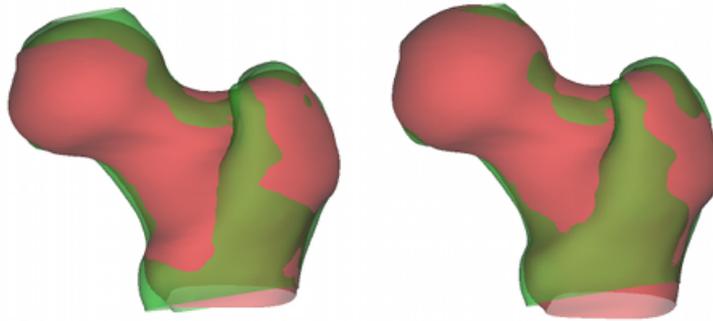


Figure 25: Example of a registration failure (*left*) that is fixed using the method of ignoring rim matches (*right*). Note the alignment of the lesser trochanters in each case.

The *normal-shooting* and *normal-weighted* matching methods (§5.3 and §5.4) are able to further reduce the number of registration failures, with the *normal-weighted* method being especially effective. This can be understood from Figure 14, since the problem of correctly aligning the lesser trochanters is closely related to case B in that diagram. The methods are not as effective as the use of distinguished points at reducing the registration failures (it was also noted that the distinguished point method gave subjectively better registrations for a number of the problematic surfaces, even when the registration was regarded as a success using both techniques). However they do have the important advantage of being more robust to ‘unusual’ surface shapes, and so do not suffer from complete failures in the way that the distinguished point method does.

The *normal-weighted* method also performed well on the synthetic data, giving a fairly significant reduction in ground truth error relative to the simple closest point matching scheme. It is likely the primary cause of this improvement is the superior global alignment. However Figure 20 shows that the scheme also has a tendency to lead to increased warping in the local stage. This could be because it tends to overfit to small scale features in the surface (like those in Figure 14B) when choosing matching points, causing the transformation to distort the surface to match these correspondences. This is not a problem with the synthetic surfaces because there are no such small scale features on them.

By contrast, the *normal-shooting* method gives a small improvement over the closest point scheme in terms of the warping of the surfaces. This is probably due to its behaviour in cases such as those illustrated in Figure 14A and 14C. The performance of this method on the synthetic data was not significantly different from that of the closest point matching.

Based on these findings, a sensible registration process could use the *normal-weighted* point matching scheme for the global alignment only. This would bring the large, important features of the surface into approximate alignment but not overfit to the smaller features. The local matching stage could then use the *normal-shooting* method (for its slightly better warping behaviour) or the closest point method (for simplicity).

Both the *normal-shooting* and *normal-weighted* methods are more computationally expensive than using simple closest points because they entail a wider search through the spatial hash look-up table. This effect increases as the weight given to the normal part of the cost function increases. However, this does not have a significant impact on the overall registration time, which is governed primarily by the nature of the optimisation in the transformation stage.

Investigation of some further methods of point matching (for example those described in §5.1) could be the focus of future work.

## 7.4 Conformal Flattening Method

For those surfaces where the conformal flattening method of §5.6 was found to work, the matches obtained from it were found to be intuitively good. For example, see Figure 26, which is a visualisation of some salient point matches found by the technique.

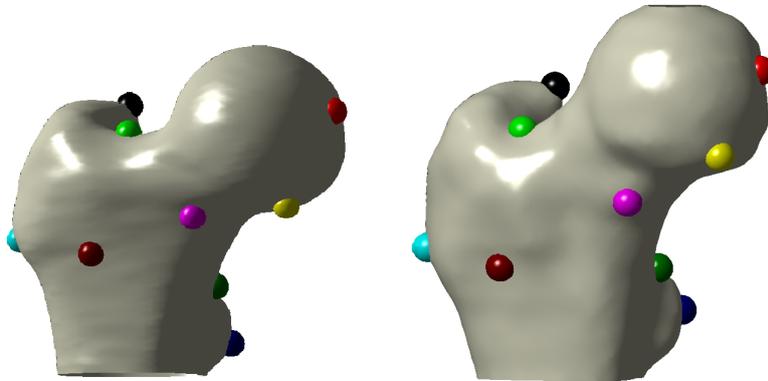


Figure 26: Examples of point matches obtained using the conformal flattening method to register the canonical femur (*left*) to a target femur (*right*). This image was produced using the Boyer et al. MATLAB implementation.

Despite this initial promise, the warping results obtained using the method are very poor compared to the other experiments. This is understandable given that there is no reason why the underlying method would keep planar points planar. The technique may just be poorly suited to our problem because it is an over-simplification to suggest that the variation in head-neck angle approximately preserves geometric distances, and there is much variation between femurs that is clearly not of this nature. Another problem with the process is that it is far more computationally expensive than the other methods considered.

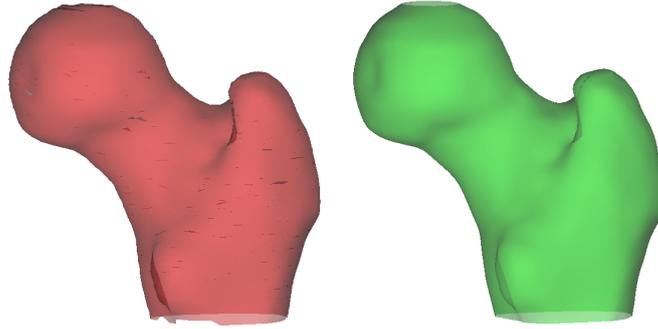


Figure 27: The canonical surface (*left*) after registration to a target surface (*right*) using the conformal flattening method. The loss of surface integrity is apparent.

More worryingly, there is a small amount of tearing and damage to the structure of the registered surfaces as shown in Figure 27. A simple solution to this could be to use the method to produce the point matches, and then find a transformation to minimise the distance between matched points. In this way the inherent constraints of chosen transformation would still apply.

More work is needed to really understand this method, whether the results obtained from it are meaningful for the registration of femurs, and why it fails completely on some surfaces. Further work could also use specific knowledge about femurs to guide the process. For example, instead of randomly sampling three points from the surface and voting for the best transformation, three meaningful distinguished points could be found and used. This may be able to reduce the occurrence of nonsensical registrations, and could make the process significantly more computationally efficient.

## 7.5 Distortion Metric

The chosen warping metric, as described in §3.2, is very simple to compute and its results have been observed to correlate with the warping observed in visualisations. However, it does not directly address the issue of what deformation is necessary in order to match the surfaces, and what deformation is superfluous. Furthermore, it only considers deformation of a small set of points on the surface (those that lie on one of the chosen contours). Therefore, conclusions drawn from warping measurements should be treated with caution.

Further work could consider improving the method used for quantifying the warping undergone by surfaces. Perhaps the method should consider directions in which points can move relative to their neighbours without affecting the ICP cost function. For example, if the shaft is considered to be approximately cylindrical, then vertical motion of points on the shaft does not effect the cost function and should therefore be considered ‘unnecessary’. Such an approach could use the differential geometry of the surface, and in particular the principal directions. Torsello et al. consider the number of similar points around a given point as an indicator of its relevance [38], though they use it for different purposes.

## 7.6 Synthetic Data and Local Minima

Figure 22 shows that none of the registration processes considered here are able to come very close to the ‘best’ registration if corresponding points are known. This shows, perhaps unsurprisingly, that the ICP cost function is a poor approximation to the true correspondence error between two surfaces. It also shows that the affine/B-spline, and affine/locally affine combinations are in fact quite capable of modelling the variation in head-neck angle and head length in principle, but are just not able to do so within the ICP framework.

It is also interesting to note that the ‘exact’ matching gives not only lower ground truth error, but also lower registration error in the case of the B-spline and locally affine transformations. This suggests that the optimisation is finding local minima of the ICP cost function, rather than the true global minimum.

Further investigation into local minima problems on the synthetic data using random parameter initialisations revealed that both the global and affine stages have a large number of local minima, and that the outcome is very sensitive to the initialisation. It is very important to find a good minimum for the global stage (in terms of ground truth error) in order to achieve low ground truth error at the end of the registration. Often a local minimum that is better than the one obtained from the existing ‘rough alignment’ (in terms of ground truth error) can be found by simply trying a few random initialisations similar to the rough alignment parameters. The same is true of random initialisation of the B-spline control points about the regularly spaced grid. Furthermore, manually choosing sensible initialisations often leads to minima with lower ground truth error (for example, manually moving B-spline control points to approximately match the head-neck angle before running the optimisation). This opens up the possibility of achieving better registrations by automatically choosing better initialisations. Such techniques could work, for example, by aligning the *skeletons* of the surfaces [39, 40].

The extent to which these findings apply to real data is unclear, since there is no ‘ground truth’ measurement in these cases. Some observations suggest that local minima are less of a problem in real data than in the synthetic data, especially for the global stage, because of the smooth, uniform nature of the synthetic surfaces and lack of lesser and greater trochanters to limit the number of suitable registrations. Choosing sensible initialisations is more difficult in the case of real data because of the need to match other features, such as the trochanters, as well as the head and neck. Nevertheless, using simple synthetic data in this way is a useful process for evaluating registration processes.

## 8 Conclusions

This project has investigated a number of different techniques for the registration of femur surfaces. Firstly, the problem of *registration failures* was addressed, and it was found that the selection of particular *distinguished points* at the tips of the two trochanters is able to dramatically reduce the number of failures, but is not robust to very unusually shaped surfaces. The use of *differential geometry* (in particular *surface curvatures*) is useful for the selection of such points.

Secondly, the unnecessary *warping* of surfaces during transformations was investigated. A simple quantification of warping based on the transformed positions of coplanar points was developed. It was found that successfully modelling variations in the head-neck angle without affecting the rest of the surface is important for avoiding such unwanted distortion, and that this is difficult to achieve using the ICP framework. Imposing constraints on the transformations was shown to have unwanted side effects, such as increasing the warping elsewhere in the surface. A *locally affine* transformation was found to have advantages over a B-spline transformation due to its more localised nature.

If used appropriately, some alternative methods for choosing matched point pairs using surface normal information and ignoring matches on the rims of surfaces were found to offer small improvements in terms of reducing registration failures and surface warping.

A sophisticated point matching scheme based on *conformal flattening* of surfaces to the complex plane was evaluated, and found to perform very poorly due to excessive warping of the registered surfaces.

It was also found that the use of synthetic data with known *ground truth* correspondence is a useful tool for evaluating registration processes. In particular, this highlights the problems of local minima in optimisation schemes and the need for good initialisation of transformation parameters.

Overall, the registration of femur surfaces is a subtle and difficult problem, and there is a large variety of techniques drawn from different areas of literature that may be used to solve it. This project has evaluated a small number of these and found some that improve the quality of the registration process. A number of possible avenues for further work have been suggested.

## Acknowledgements

I would like to thank Andrew Gee for his close supervision of this project, his frequent discussions, advice, encouragement and computing assistance. Andrew wrote the *wxRegSurf* software, and provided the real and synthetic data for experiments.

I would also like to thank Chenzi Xu for her encouragement, her suggestions for the preparation of figures, and for proofreading this report.

## References

- [1] J.J.W. Roche, R.T. Wenn, O. Sahota, and C.G. Moran. Effect of comorbidities and postoperative complications on mortality after hip fracture in elderly people: prospective observational cohort study. *British Medical Journal*, 331:1374, 2005.
- [2] C. Currie, M. Partridge, F. Plant, J. Roberts, R. Wakeman, and A. Williams. The National Hip Fracture Database National Report. Technical report, The National Hip Fracture Database, 2012.
- [3] G. Holzer, G. von Skrbensky, L.A. Holzer, and W. Pichl. Hip fractures and the contribution of cortical versus trabecular bone to femoral neck strength. *Journal of Bone and Mineral Research*, 24(3):468–474, 2009.
- [4] K.E.S. Poole, G.M. Treece, P.M. Mayhew, J. Vaculík, P. Dungal, M. Horák, J.J. Štěpán, and A.H. Gee. Cortical thickness mapping to identify focal osteoporosis in patients with hip fracture. *PLoS ONE*, 7(6), 2012.
- [5] K.E.S. Poole, G.M. Treece, G.R. Ridgway, P.M. Mayhew, and J. Borggrefe. Targeted regeneration of bone in the osteoporotic human femur. *PLoS ONE*, 6(1), 2011.
- [6] G.M. Treece, A.H. Gee, P.M. Mayhew, and K.E.S. Poole. High resolution cortical bone thickness measurement from clinical CT data. *Medical Image Analysis*, 14:276–290, 2010.
- [7] M.A. Audette, F.P. Ferrie, and T.M. Peters. An algorithmic overview of surface registration techniques for medical imaging. *Medical Imaging Analysis*, 4:201–217, 2000.
- [8] P.J. Besl and H. McKay. A method for the registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [9] Y. Chen and G. Medioni. Object modeling and registration of multiple range images. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1991.

- [10] D. Rueckert, P. Aljabar, R.A. Heckemann, J.V. Hajnal, and A. Hammers. Diffeomorphic registration using B-splines. *Medical Imaging Computing and Computer Assisted Intervention*, 9(2):702–709, 2006.
- [11] P.J. Besl and R.C. Jain. Invariant surface characteristics for 3D object recognition in range images. *Computer Vision, Graphics, and Image Processing*, 33:33–80, 1986.
- [12] M. Brady, J. Ponce, A. Yuille, and T. Asada. Describing surfaces. In *Computer Vision, Graphics, and Image Processing*, pages 1–28. MIT Press, 1985.
- [13] A.P. Mangan and R.T. Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5:308–321, 1999.
- [14] P.J. Besl and R.C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:167–192, 1988.
- [15] H. Yamauchi, S. Gunhold, R. Zayer, and H-P. Seidel. Mesh segmentation driven by Gaussian curvature. *The Visual Computer*, 21:659–668, 2005.
- [16] J-P. Thirion. The extremal mesh and the understanding of 3D surfaces. *International Journal of Computer Vision*, 19(2):115–128, 1996.
- [17] G. Subsol, J-P. Thirion, and N. Ayache. Non-rigid registration for building 3D atlases. In *Proceedings of the 12th IAPR International Conference on Computer Vision and Image Processing*, volume 1, pages 576–578, 1994.
- [18] J-P. Thirion. Extremal points: definition and application to 3D image registration. In *Proceedings of The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 578–592, 1994.
- [19] D.B. Goldgof, H. Lee, and T.S. Huang. Feature extraction and terrain matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 899–904, 1988.
- [20] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 373–380, 2009.
- [21] C.S. Chua and R. Jarvis. Point signatures: a new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.

- [22] P. Csákány and A.M. Wallace. Computation of local differential parameters on irregular meshes. In *The Mathematics of Surfaces IX*, pages 19–33. Springer-Verlag, 2000.
- [23] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. Technical report, IBM T.J. Watson Research Center, 1995.
- [24] M. Meyer, M. Desbrun, P. Schröder, and A.H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, pages 35–57, 2003.
- [25] R. Szeliski and S. Lavallée. Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. *International Journal of Computer Vision*, 18(2):171–186, 1996.
- [26] D. Rueckert, L.I. Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, and D.J. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, 1999.
- [27] F.L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3), 1989.
- [28] Z. Xie and G.E. Farin. Image registration using hierarchical B-splines. *IEEE Transactions on Visualization and Computer Graphics*, 10(1), 2004.
- [29] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of smooth surfaces. Technical Report 2220, INRIA, 1994.
- [30] D. Münch, B. Combès, and S. Prima. A modified ICP algorithm for normal-guided surface registration. In *Proceedings of SPIE*, volume 7263, 2010.
- [31] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the International Conference on 3-D Digital Imaging and Modelling*, 2001.
- [32] Q-X. Huang, B. Adams, M. Wicke, and L.J. Guibas. Non-rigid registration under isometric deformations. *Computer Graphics Forum*, 27(5):1449–1457, 2008.
- [33] D. Anguelov, P. Srinivasan, H-C. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Information Processing Systems*, volume 17, 2004.
- [34] X. Gu and B.C. Vemuri. Matching 3D shapes using 2D conformal representations. In *Medical Imaging and Computer Aided Intervention*, pages 771–780, 2004.

- [35] Y. Lipman and T. Funkhouser. Möbius voting for surface correspondence. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), 2009.
- [36] D.M Boyer, Y. Lipman, E. St. Clair, J. Puente, B.A. Patel, T.A. Funkhouser, J. Jernvall, and I. Daubechies. Algorithms to automatically quantify the geometric similarity of anatomical surfaces. *Proceedings of the National Academy of Sciences of the United States of America*, 108(45):18221–18226, 2011.
- [37] K. Turkowski. Transformations of surface normal vectors with applications to three dimensional computer graphics. Technical Report 22, Apple Computer, Inc. 1990.
- [38] A. Torsello, E. Rodolà, and A. Albarelli. Sampling relevant points for surface registration. In *Proceedings of the 2011 IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 290–295, 2011.
- [39] T.K. Dey and J. Sun. Defining and computing curve-skeletons with medial geodesic function. In *Eurographics Symposium on Geometry Processing*, 2006.
- [40] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su. Point cloud skeletons via Laplacian based contraction. In *Shape Modeling International Conference*, pages 187–197, 2010.

# Appendix

## A Risk Assessment Retrospective

The practical parts of this project were entirely computer-based. The original risk assessment identified repetitive strain injury and other health problems related to extended use of computers, which retrospectively seems appropriate. I was mindful of these problems during the project and took simple steps to alleviate them. Fortunately, no such problems did occur.