Faculteit Wetenschappen
Departement Fysica

Surface and image-based registration methods with statistical modeling for biomedical applications

---

Oppervlak- en beeld-gebaseerde registratie methodes met statistische modellering voor biomedische toepassingen

Proefschrift voorgelegd tot het behalen van de graad van

Doctor in de Wetenschappen: Fysica

aan de Universiteit Antwerpen, te verdedigen door

Jeroen VAN HOUTTE

Promotoren:
Prof. Dr. Jan Sijbers
Prof. Dr. Toon Huysmans

Antwerpen, 2023

Doctoral jury:

Prof. Dr. Pierre Van Mechelen (University of Antwerp)
Prof. Dr. Peter Aerts (University of Antwerp)
Prof. Dr. Jan Sijbers (University of Antwerp)
Prof. Dr. Toon Huysmans (Delft University of Technology)

External jury members:
Prof. Dr. Guoyan Zheng (Shanghai Jiao Tong University)
Prof. Dr. Emmanuel Audenaert (Ghent University Hospital)

Contact information:

✉ **Jeroen Van Houtte**
imec - Vision Lab, Dept. of Physics
University of Antwerp (CDE)
Universiteitsplein 1, Building N1.19
B-2610 Wilrijk, Antwerpen, Belgium
☎ +32 499 75 62 54
jeroen.vanhoutte@uantwerpen.be
https://visielab.uantwerpen.be/people/jeroen-vanhoutte

"You can have data without information,
but you cannot have information without data."

DANIEL KEYS MORAN

Contents

Summary

Over the past decade, digital data generation and collection has become increasingly important in biomedicine. Surgeons heavily rely on biomedical data for diagnoses, pre-operative planning, follow-up, etc on a daily basis. With advancing imaging technologies, large amounts of data have become available in different modalities, such as optical surface scans and images. It is the challenge of this era to exploit the information in this data in order to expand our knowledge in biomedicine and to improve our healthcare system. Learning from large collections of data can help us in automating diagnoses that would be purely based on data, in contrast to subjective decisions that are based on the experience of a surgeon. The knowledge of shape variability in a large dataset, as another example, can be a guide for product development. Letting a computer "understand" images based on past examples enables computer-assisted robotic surgeries.

This thesis contributes to the data-driven solutions in biomedicine. On a first level, we present a framework to combine the shape information of different patients into one digital model. Such statistical model can be built on images or surface models. On a second level, these digital models serve as prior knowledge for computers to automatically "understand" new data.

A crucial step on both, the modeling and the application level, is the anatomical alignment of data. This step, known as *registration*, is important for, for example, tracking of a moving body over time or the digital mapping of an implant onto a patient. While finding corresponding points between different data is easy for human eyes, this problem remains a challenging task to computers.

The manuscript is divided into three parts. Part I provides an introduction to the necessary concepts, being: geometry and image processing in biomedicine, X-ray imaging and deep-learning (DL). Part II and III include the contributions of this PhD to surface and image-based registration methods, respectively.

Contributions to surface registration of articulating bodies

Part II of the thesis focuses on shape modeling of articulating objects. Such models are important to facilitate biomechanical simulations that help us understand the development and treatment of certain pathologies. In a clinical context, they can help in detecting motion abnormalities and thereby preventing injuries.

One way to study motion is by tracking reflective markers attached to the patient at certain anatomical locations. The markers, however, only provide the location of a sparse set of points over time. This sparse data does not provide information on the level of the bone surfaces themselves, such as the knee joint space distance for

example. In chapter 4 we show how a person-specific shape model can be registered to such set of markers to provide such surface-based measurements.

Person-specific models are, however, not always available and can be costly and time consuming to acquire. For such cases, we have built generic statistical models of articulating structures, which describe the statistical variations in shape in a certain population while maintaining the possibility to be articulated into different poses. With such statistical models any individual in the population can be described up to a certain accuracy. Hence, the acquisition of person-specific models is no longer required.

Two different articulating statistical shape models have been built to illustrate its usefulness in different application fields. Chapter 5 presents a statistical shape model of the human hand that can be used to automatically design a splint based on a low quality 3D-scan. Chapter 6 presents a statistical shape model of the horse limb for veterinary applications.

The ability of statistical models to describe many individuals in a certain population makes them also very interesting for deep-learning models (see introduction chapter 3). Training these models requires large labeled datasets, which are time-consuming to acquire in practice. From a statistical model, however, synthetic data can easily be generated, along with ground-truth labels. This has been illustrated in chapter 7 for the training of a 2D-image segmentation network which required ground-truth labeling of the structures.

Contributions to DL-based 2D/3D image registration

Part III of the thesis focuses on solving a specific registration problem in X-ray imaging (see introduction chapter 2), through deep-learning. As opposed to part II of the thesis we use an image-representation of the patient instead of a surface-representation.

X-ray imaging or radiography is the most common imaging procedure for many orthopedic interventions thanks to its ability to visualize internal structures with a relatively low radiation dose and low acquisition cost. However, interpretation from two-dimensional (2D) radiographs can be hampered by overlapping structures, magnification effects and the patient's positioning. To avoid the difficulties associated with 2D projections, we developed two methods to register a 3D model to a pair of radiographs. The registered model enables a 3D-interpretation, while keeping the benefits of RX over CT, in terms of costs and radiation dose.

The first solution for this 2D/3D registration problem is presented in chapter 8. This model constraints the possible solutions through a prior statistical model, which encodes the possible shapes across a population. Instead of reconstructing a 3D volume, the network predicts the weights of the statistical model.

Chapter 9 presents a second solution. This registration network regresses a dense 3D deformation field, without a statistical prior model. The deformation field warps an atlas image such that the forward projection of the warped atlas matches the input 2D radiographs.

Samenvatting

Digitale data generatie en verwerving is over het voorbije decennium steeds belangrijker geworden in de biomedische wereld. Artsen maken veelvuldig gebruik van biomedische data voor diagnoses, pre-operatieve planning, opvolging, enz. Met de technologische vooruitgang in beeldvormingstechnieken, is de beschikbaarheid van data in verschillende modaliteiten, zoals optische oppervlak scans en beelden, sterk toegenomen. Het is onze uitdaging om zo veel mogelijk informatie uit deze data te halen om onze kennis in biomedische wetenschappen uit te breiden en om onze gezondheidszorg te verbeteren. Door te leren uit grote collecties van data, kunnen we diagnoses automatiseren die enkel berust zijn op data, in tegenstelling tot subjectieve beslissingen die gebaseerd zijn op de ervaring van een arts. De kennis van vorm variaties in een grote dataset kan gebruikt worden in product ontwikkeling bijvoorbeeld. Op basis van voorgaande voorbeelden, kan een computer nieuwe beelden automatisch "begrijpen", wat computer-geassisteerde operaties met robotica mogelijk maakt.

Dit doctoraatsonderzoek draagt bij tot data-gedreven oplossingen voor biomedische toepassingen. Op een eerste niveau, ontwikkelen we een manier om de vorm-informatie van verschillende patiënten te combineren in één digitaal model. Een dergelijk model kan gegenereerd worden op basis van beelden of oppervlak modellen. Op een tweede niveau worden deze modellen gebruikt als voorkennis om computers nieuwe beelden te laten "begrijpen".

Een cruciale stap in beiden, de modellering en de toepassing, is het anatomisch aligneren van data. Deze stap staat bekend als *registratie*, en is bijvoorbeeld belangrijk voor het volgen van een bewegend lichaam over de tijd of het digitaal overbrengen van een implantaat op een patiënt. Terwijl het bepalen van correspondenties tussen verschillende data een gemakkelijk probleem lijkt voor het menselijk oog, is dit voor computers nog steeds een uitdaging.

De thesis is opgedeeld in drie delen. Deel I biedt een inleiding over oppervlak- en beeldverwerking in biomedische toepassingen, X-stralen beeldvorming en "deep learning" (DL). Deel II en III bevatten respectievelijk de bijdragen van dit doctoraatsonderzoek tot de domeinen van oppervlak- en beeld-registratie.

Bijdrage tot oppervlak registratie van articulerende lichamen

Deel II van de thesis richt zich tot vorm modellen van articulerende objecten. Zulke modellen zijn bijvoorbeeld belangrijk om biomechanische simulaties mogelijk te maken, die op hun beurt onze kennis over de ontwikkeling en genezing van pathologiën verbeteren. In een klinische context, kunnen ze gebruikt worden voor het detecteren van afwijkende bewegingspatronen en kunnen daarbij letsels voorkomen.

De voortbeweging van individuen wordt vaak bestudeerd aan de hand van reflecterende

markers op de huid of botten waarvan de posities wordt bijgehouden in functie van de tijd. Dit geeft echter enkel informatie over de beweging van deze enkele punten. Deze beperkte datapunten geven geen directe informatie over, bijvoorbeeld, de afstand tussen botten in een knie gewricht. In hoofdstuk 4 tonen we hoe een vorm model van een patiënt kan geregistreerd worden aan deze set van markers om oppervlak-gebaseerde metingen mogelijk te maken.

Een oppervlak model van de patiënt zelf is echter niet altijd beschikbaar. In zo'n situatie kan gebruik worden gemaakt van generisch articulerende statistische modellen die zowel de vormvariatie en pose-veranderingen beschrijven in een bepaalde populatie. Elk individu uit de populatie kan tot een bepaalde nauwkeurigheid worden beschreven met een dergelijk model zonder dat een afzonderlijke opmeting moet gebeuren voor dat individu.

In dit doctoraatsonderzoek werden twee verschillende statistische modellen ontwikkeld voor twee verschillende toepassingen. Hoofdstuk 5 beschrijft een statistisch model van een menselijke hand voor het geautomatiseerd ontwikkelen van orthopedische spalken op basis van lage kwaliteit scans. Hoofdstuk 6 beschrijft een statistisch model van een paardenledemaat voor veterinaire toepassingen.

De mogelijkheid om uit een statistisch model verschillende synthetische individuen te construeren, maakt dit soort modellen ook uiterst interessant voor "deep learning"-toepassingen (zie introductie hoofdstuk 3). De training van zulke modellen vereist grote datasets van gelabelde data, wat tijdrovend kan zijn om deze te verwerven. Van een statistisch model kan echter synthetische data worden gegenereerd, samen met de labels. Dit werd geïllustreerd in hoofdstuk 7 voor een 2D segmentatie netwerk dat voor de training de labels van verschillende structuren vereiste.

Bijdrage tot DL-gebaseerde beeld-registratie

Deel III van de thesis focust op een specifiek registratie probleem in X-stralen beeldvorming (zie introductie hoofdstuk 2), dat wordt opgelost aan de hand van "deep-learning". In tegenstelling tot deel II van de thesis, wordt hier gebruik gemaakt van beelden in plaats van oppervlak modellen als digitale representatie van de patiënt.

X-stralen beeldvorming of radiografie is de meest gebruikte beeldvormingsmethode voor orthopedische interventies vanwege de mogelijkheid om interne delen te beeldvormen tegen relatief lage kost en lage stralingsdosis. De interpretatie van een 2D radiografie beeld kan echter bemoeilijkt worden vanwege overlappende structuren, vergrotingseffecten en de positie van de patiënt. Om deze moeilijkheden te omzeilen werden in dit doctoraatsonderzoek twee methodes ontwikkeld om een 3D model te registreren aan de 2D data. Dit maakt een 3D interpretatie van de 2D data mogelijk terwijl de voordelen van een RX beeld ten opzichte van een CT-opname behouden blijven.

Het eerste 2D/3D registratienetwerk, beschreven in hoofdstuk 8, maakt gebruik van een statistisch vervormingsmodel. Dit model bevat de toegelaten vormvariaties en beperkt zodoende de mogelijke oplossingen. In plaats van het 3D beeld meteen te reconstrueren, voorspelt het netwerk de gewichten van het statistisch model die corresponderen met het juiste 3D beeld.

Het tweede registratiemodel, beschreven in hoofdstuk 9, schat rechtsreeks een 3D deformatie veld, zonder gebruik te maken van een statistisch model. Het geschatte vervormingsveld wordt gebruikt om een 3D atlas-beeld te vervormen zodanig dat de voorwaartse projectie ervan overeenkomt met de input 2D RX-beelden.

List of Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
4D	Four-dimensional
AAM	Active appearance model
ACL	Anterior cruciate ligament reconstruction
ACWE	Active contour model without edge
Adam	Adaptive moment estimation
AI	Artificial intelligence
ANN	Artificial neural network
AP	Anterior-posterior
ARAP	As rigid as possible
ASM	Active shape model
ASSD	Average symmetric surface distance
CAD	Computer-aided design
CAS	Computer-assisted surgery
CAUD	Caudal angle
CLM	Constrained local model
CNN	Convolutional neural network
CPU	Central processing unit
CRAN	Cranial angle
CT	Computed tomography
DIP	Distal interphalangeal joint
DL	Deep learning
DNN	Deep neural network
DRF	Digital reference frame
DRR	Digitally reconstructed radiograph
ELU	Exponential linear unit
FC	Fully connected layer
FCN	Fully convolutional network
FEM	Finite element model
FFD	Free-form deformation
FOV	Field of view
GAC	Geodesic active contour
GAN	Generative adversarial network
GPU	Graphics processing unit
GUI	Graphical user interface
HU	Hounsfield unit
ICP	Iterative closest point

IR	Infrared
ISB	International society of biomechanics
KPCA	Kernel principal component analysis
LAO	Left anterior oblique angle
LAT	Lateral
LBS	Linear blend skinning
LIDAR	Light detection and ranging
LM	Levenberg-Marquardt
MC	Monte Carlo
MCP	Metacarpophalangeal joint
ML	Machine learning
MRI	Magnetic resonance imaging
NCC	Normalised cross-correlation
NN	Neural network
NURBS	Non-uniform rational B-splines
PDM	Point distribution model
PC	Principal component
PCA	Principal component analysis
PGA	Principal geodesic analysis
PIP	Proximal interphalangeal joint
PMT	Photomultiplier tube
ProST	Projective spatial transform layer
RAO	Right anterior oblique angle
ReLU	Rectified linear unit
RGB	Red-green-blue
RX	X-ray
SDM	Statistical deformation model
SLERP	Spherical linear interpolation
SM	statistical model
SSAM	Statistical shape and appearance model
SSIM	Statistical shape and intensity model
SSIM	Structural similarity index
SSM	Statistical shape model
ST	Spatial transform
SVD	Singular value decomposition
SVM	Support-vector machine
TKA	Total knee arthroplasty

Part I

Introduction

1

Digital geometry and image processing in biomedicine

Contents

1.1 Computer-aided applications in biomedicine

In this section we introduce the biomedical applications envisioned in this thesis and how computer algorithms help in these applications. We limit ourselves to orthopaedical applications which deal with the musculoskeletal system.

Biomedical simulations

Knowing the origin and development of orthopedic pathologies is important for a proper diagnosis and treatment. Our understanding is mostly based on experience from medical practices, but advancements in computer simulations allow to investigate the underlying mechanisms more thoroughly. They offer a controlled environment in which the influence of different factors such as shape geometry, bone density, soft tissue and loading can be studied on, for example, the kinematics. They also allow to estimate quantities that can not be measured experimentally, such as cartilage stress. Statistical shape information in such simulation frameworks allows to investigate the correlation between bone shape and biomechanical function **??**.

Computer-aided diagnoses and follow-up

Computer-aided diagnosis systems assist medical practitioners in interpreting biomedical data of a patient. As they can learn from previous data and can process data faster, they can guarantee a more precise outcome. Early-stage signatures of diseases like osteoarthritis and cancer, for example, can easily be overlooked by the human eye, while their treatment would benefit from an early detection and diagnosis **??**.

Gait analysis

Abnormalities in the gait of a patient, such as motion asymmetry, can be an indication for a functional or structural disorder in the musculoskeletal system **?**. In the veterinary field, such manifestation is called lameness and is associated to reduced performance of sport animals. Diagnosing the deviating motion patterns can be very subjective when done visually, but by acquiring motion data, either indirect by measuring contact forces or direct by motion tracking, a quantitative gait analysis can be delivered. This can be done for diagnosis, for evaluation of a certain treatment, or for preventive monitoring.

Computer-assisted surgeries

Computer-assisted surgeries (CAS) refer to workflows which rely on biomedical data and computer technologies to help a surgeon, either pre-operatively (before surgery), intra-operatively (during surgery) or post-operatively (after surgery). They show advantages in terms of precision, reproducability, clinical outcome, time efficiency, safety and reduced invasiveness **???**

Pre-operative planning starts with generating a digital representation of the patient. This involves the identification of the region of interest through labeling. The labeling of components, called “segmentation”, is a laborious task in general, where computer-algorithms provide an automatic or semi-automatic solution. It is the basis for more dedicated data-processing algorithms, such as bone fracture detection for example.

The digital representation allows the surgeon to plan the surgery virtually with dedicated software programs.

During the operation, the surgeon wants to follow the pre-operative plan as accurate as possible. This requires a mapping of the virtual pre-operative plan onto the physical patient, based on intra-operative data. Based on this data, navigation systems can track the position and orientation of surgical tools and implants with respect to the patient. The possibility to track based on data allows for small incisions and limited invasiveness of the surgery. The positioning of the instruments with respect to the patient can be displayed on a screen or virtually overlaid on top of the patient by means of augmented-reality glasses **?**. The navigation helps a surgeon or robotic system to accurately manipulate the instruments and to position drilling holes or cutting planes as planned. During the procedure, live feedback can be given on the potential clinical outcome of the surgery, based on real-time biomechanical simulations. This allows, for example, to fine-tune implant positioning intra-operatively.

Post-operative data can be compared to pre-operative data to evaluate the success of the operation. By aligning data acquired at different times, the progression of a disease or treatment can be followed up.

Product development

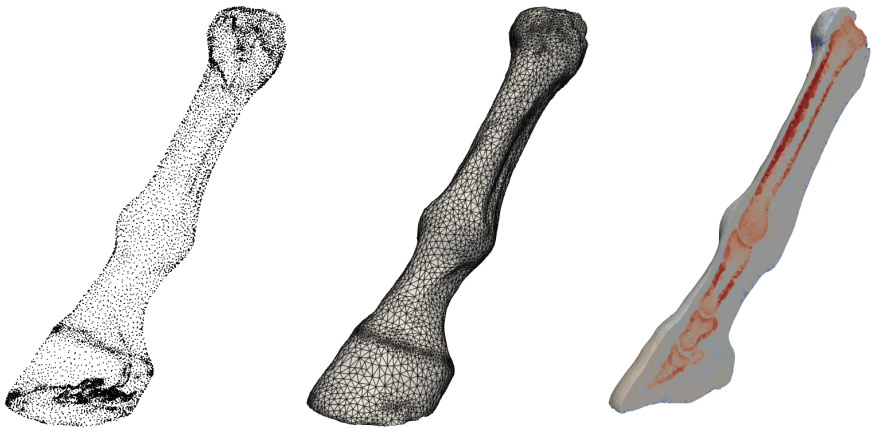
Patient-specific digital models allow for the design of personalised orthopedic implants that are customised to the patients anatomy. In contrast to off-the-shelf products, they improve the medical outcome and reduce the number of needed revisions **?**.

1.2 Biomedical data representations

Physical objects around us are defined by a continuous surface, characterised by continuous properties like curvature. Computers however need a discrete representation of those objects to understand a "shape". The actual shape X is therefore sampled by a set of points, which can either be regularly distributed like in an image or irregular like in a mesh. In case of images, we speak of pixels or voxels for the 2D or 3D case, respectively. Common modalities of medical images include radiographs and computed tomography (CT), as introduced in chapter 2, but also magnetic resonance imaging (MRI) and ultrasound. In this section we will focus on the irregular shape representations and discuss the different types and their acquisition methods.

1.2.1 Geometric shape representations

- Point clouds are a set of points distributed across the surface, without necessarily embodying feature locations. It is the simplest and most generic representation. An example is shown in Figure 1.1a.
- Polygonal surface meshes are graphs, embedded in Euclidean space. They are characterised by a set of points connected by edges, such that the edges span triangular or quadrilateral cells. As illustrated in Figure 1.1b, the polygons form a piecewise planar approximation to the actual shape. The Delaunay triangulation of a set of points maximises the smallest occurring angle between edges.



(a) Point cloud (b) Surface mesh (c) Tetrahedral model

Figure 1.1: Different discrete geometry representations of a horse leg.

- Volumetric tetrahedral models use tetrahedral cells to fill up the object. A cross-section of such model is shown in Figure 1.1c. A point $\boldsymbol{x} \in \mathbb{R}^3$ inside a single tetrahedron $T = [\boldsymbol{v_0}, \boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}]$ can be described by its barycentric coordinates $\boldsymbol{u} = [u_0, u_1, u_2, u_3]$ relative to T, such that $\boldsymbol{x} = u_0\boldsymbol{v_0} + u_1\boldsymbol{v_1} + u_2\boldsymbol{v_2} + u_3\boldsymbol{v_3}$. This representation has the benefit that volumetric information can be stored, in terms of trivariate Bernstein basis polynomials for example:

$$B^d_{\boldsymbol{k}}(\boldsymbol{u}) = \binom{d}{\boldsymbol{k}} u_0^{k_0} u_1^{k_1} u_2^{k_2} u_3^{k_3} \tag{1.1}$$

with the binomial coefficient $\binom{d}{\boldsymbol{k}} = \frac{d!}{k_0!k_1!k_2!k_3!}$. A Bernstein polynomial is a linear combination of Bernstein basis polynomials:

$$f^j(\boldsymbol{u}) = \sum_{|\boldsymbol{k}|=d} \beta^j_{\boldsymbol{k}} B^d_{\boldsymbol{k}}(\boldsymbol{u}) \tag{1.2}$$

with $\beta^j_{\boldsymbol{k}}$ the Bernstein coefficients. The sum runs over all combinations of k_0, k_1, k_2 and k_3, for which the sum $\sum_{i=0}^{3} k_i$ equals the degree d.

- Medial models describe a surface by its center-line and the radii along this line. In the original work of Blum et al., each point on the medial line represents the center of the largest possible inscribing sphere to the object, such that each sphere is bitangent to the objects boundary **?**. While these models derive a continuous medial axis from the boundary, m-rep models imply the boundary from a mesh of medial atoms **?**. The medial atoms are located at the sphere centers, and have two equally-sized spokes normal to the object boundary. The end of each spoke thus gives the position and normal at the two intersection points of the sphere with the surface.
- Surfaces with spherical topology can be parameterised such that a pair of polar coordinates (θ, ϕ) maps to a surface coordinate $\boldsymbol{v}$ by the following three

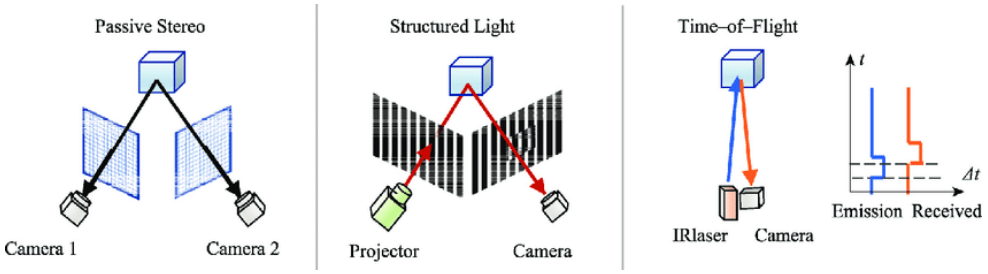


Figure 1.2: Different capturing techniques for 3D objects **?**.

coordinate functions:

$$\boldsymbol{v}(\theta,\phi) = \begin{pmatrix} x(\theta,\phi) \\ y(\theta,\phi) \\ z(\theta,\phi) \end{pmatrix} \tag{1.3}$$

These coordinate functions can be decomposed over a set of basis functions, such as B-splines, wavelets or spherical harmonics **?**.

- Non-uniform rational B-splines (NURBS) models control the shape by a limited number of control points P_{ij} in each of two directions (u, v). A shape vertex position is given by an interpolation of neighboring control points:

$$s(u,v) = \frac{\sum_{i=0}^{N_u-1}\sum_{j=0}^{N_v-1} w_{ij} B_i^n(u) B_j^n(v) P_{ij}}{\sum_{i=0}^{N_u-1}\sum_{j=0}^{N_v-1} w_{ij} B_i^n(u) B_j^n(v)} \tag{1.4}$$

with B_i^n recursively-defined B-spline basis functions of degree n.

- A levelset method represents a closed surface boundary $\mathcal{C}$ implicitly by the zero-levelset of the levelset-function ϕ:

$$\mathcal{C} = \{\boldsymbol{x} \in \Omega | \phi(\boldsymbol{x}) = 0\} \tag{1.5}$$

The interior of the shape is made up of all the points for which ϕ is positive. In numerical computations, the levelset-function ϕ is often set to the signed euclidean distance function to the surface.

1.2.2 Shape acquisitions

From image to surface model

Extraction of an object as surface model from a 3D image first requires segmenting out the object of interest from the image. A segmentation map is a binary classification of voxels telling which voxels belong to the object or to the background. A binary voxelised representation of the object can be converted into a polygonal surface model by means of a marching cube algorithm **?**. For every set of eight neighboring voxels at the boundary it proposes a set of polygons based on their segmentation values, to approximate the surface passing through the voxels. There are 2^8 possible cases, each having its own pre-defined polygonal partitioning. Finally, vertices sharing the same edge are interpolated to obtain a closed mesh.

The marching cube algorithm does not necessarily result in a uniform mesh and its output mesh can be very complex due to the high resolution of the input 3D image.

Many geometry remeshing libraries are available to further improve the mesh quality by coarsening the mesh, smoothening the surface, improving triangle aspect ratios, removing mesh artefacts, etc **?**. Opposed to uniform remeshing, making the mesh resolution adaptive to high curvature regions can be of interest to preserve certain features, while reducing the total amount of vertices **?**.

Passive stereo vision

Stereo vision systems mimic the human binocular vision by looking at a scene from two slightly different perspectives, as illustrated in Figure 1.2. The spatial shift between corresponding image points is encoded in the disparity map, from which the relative depth of the pixels can be calculated. It is a passive technique, meaning that it does not illuminate the scene itself. This means that the object needs to be sufficiently illuminated by ambient light and that it needs to have some texture or features.

Optical probe

Optical tracking systems use stereoscopic vision to track the 3D position of retroreflective markers that are attached to a patient or to a surgical instrument. The optical system consists of, at least, two infrared (IR) illuminators and two camera sensors that detect the reflected IR light from the markers. By arranging a set of several markers into a specific configuration on a surgical instrument, the position sensor is able to calculate the position and orientation of that instrument, in the form of a rigid transformation. A commercial example is Polaris Vega optical tracking system from Northern Digital Inc. (NDI) **?**.

Structured light scanning

A structured light 3D scanner, depicted in Figure 1.2, can be seen as an active stereo-vision system, as it actively projects a temporally or spatially 2D light pattern onto the scene in order to ease the matching between corresponding points. One or two cameras capture the reflected light, and based on the deformation of the pattern, the depth can be calculated. Examples of brands that manufacture commercial structured light scanners are Artec and 3DMD.

Time-of-flight camera

Time-of-flight (TOF) cameras, illustrated in Figure 1.2, determine the distance to an object by measuring the time it takes for a laser pulse to travel back and forth between the camera and the object. Short-pulse TOF systems, like LIDAR, use short infrared laser or led light pulses. Continuous-wave TOF cameras emit modulated light pulse and measure the phase shift between the emitted and reflected wave to determine the distance.

1.3 Registration problem

Registration is the process of aligning two or more datasets with each other, such that corresponding features on the different datasets overlap with each other. The data to be aligned can have been acquired at different times or by different imaging sensors,

or from different patients or view-points. In this section we give some examples of registration in biomedical problems, before discussing the registration problem from a mathematical point of view, for the case of surface meshes and images.

1.3.1 Applications of registration in biomedicine

Registration plays an important role in the pre-processing pipeline of many biomedical procedures. By bringing different data into the same coordinate system, registration can help medical practitioners to save time in interpreting the data correctly.

Intra-person registration is valuable to monitor disease progression in longitudinal studies. During the COVID pandemic, for example, it could have been used to monitor the progression of different diseased lung regions over time by subtracting two registered lung CT-images with each other to highlight density changes for different infected regions **?**. Similarly, it can be used in cancer treatment to track the size of different tumors over time and to adjust radiation therapy accordingly **?**.

Shape changes on a shorter timescale, such as a beating heart or respiratory motion, can be captured by imaging systems like 4D-CT, where temporal registration is capable of building a temporal model and identifying pathological deformations **?**. Having a respiratory model of a patient's lung enables the prediction of tumor motion and deformation, which can be used for motion compensation in radiation therapy to avoid damaging healthy tissue **?**. For non-cyclic motion, like joint motion, 4D-CT acquisitions result in partial data only. In that case, a static 3D CT scan can be registered to the partial 4D-CT data to yield the motion of the complete 3D shapes over time **?**.

Often data of different modalities are acquired from the same patient as they showcase different characteristic information, and are complementary to each other. Fusion of those different modalities, referred to as multimodal registration, provides the medical practitioners a more complete view on the patient **?**. The inherent different appearance of the source and target make multimodal registration a challenging task.

Next to surgical planning and diagnosis, multi-modal registration is particularly important for computer-assisted-surgeries (CAS), where a pre-operative image, like CT or MRI, needs to be registered to intra-operative data, to enable image-guided navigation and robotic positioning. Commercial medical robotic systems use optically-track-able markers as inter-operative data such that the medical plan can be matched onto the patient **?**. As markers are invasive, research is being done on matching the pre-operative data onto ultrasound images **?** or to 2D fluoroscopy images **?**.

Opposed to registration of datasets of the same patient, inter-person registration is done between two or more different subjects, often a patient and a reference atlas or template. This facilitates atlas-based segmentation or landmarking, where the atlas segmentation map and annotated features are mapped to the new unsegmented data by the registration function **?**.

Registration between a patient and an atlas also enables automatic implant design, by warping a standardised atlas-implant with the registration transformation to the patient **?**. Such automated procedure saves time for the clinicians and ensures optimal fitting of the implant to the patient. If the fabrication of personalised implants is

not feasible, registration methods can still be used to evaluate the fitting quality of off-the-shelf implants by rigidly registering the implant to different individual shapes **?**.

1.3.2 General solution to registration

A registration method seeks an optimal spatial transformation that aligns the moving source data $\mathcal{M}$ with the fixed target data $\mathcal{F}$. Classically, this is treated as an optimisation problem, consisting of three essential parts: a transformation T that drives the source data, a similarity metric that evaluates the quality of the alignment and an optimizer which minimizes the metric by tuning the transformation parameters.

Possible transformations fall into two main categories: affine and deformable transformations. The former category accounts for translation, rotation, scaling and shear. The latter is used to accommodate local deformations between the source and target, which arise because of inter-person shape variations or temporal deformations. The transformation T is typically parameterised by a set of parameters β.

The suitability of the transformation is quantified by a certain energy function or metric E, which measures the overlap between the target data and the transformed source. For registration of discrete geometries, the metric can be an euclidean point-to-point distance or point-to-surface distance. For image registration, popular metrics include the sum of squared differences (SSD), normalised cross-correlation (NCC) and mutual information metric (MI).

The goal of the optimiser is to find the transformation parameters $\hat{\beta}$ that minimise the energy function E:

$$\hat{\beta} = \arg\min_{\beta} E(\mathcal{F}, \mathcal{M}(T(\beta))) \tag{1.6}$$

Starting from an initial parameter guess β_0, the optimiser tries to find the (global) minimum of the energy function by iteratively updating the transformation parameters. The gradient descent optimiser, the simplest optimisation method, updates the transformation parameters in the opposite direction as the gradient of the energy function $\frac{\partial E}{\partial \beta}(\beta_i)$:

$$\beta_{i+1} = \beta_i - \alpha \frac{\partial E}{\partial \beta}(\beta_i), \tag{1.7}$$

with α a user-defined step size or learning rate. If α is too small, it will take a long time for the optimiser to converge if the energy surface is flat. If α is too high, however, there is a risk that the optimiser will never reach the global minimum and will jump over it each time in case of a steep energy valley. More sophisticated methods can modify the learning rate based on the higher order curvature of the energy surface. A Levenberg-Marquard optimisation scheme, for example, follows the gradient-descent approach far from the minimum, but closer to the minimum, it will gradually move to a Gauss-Newton method. Hereby the energy function is being approximated locally by a quadritic function, for which the minimum acts as the next guess of the global energy minimum.

1.3.3 Surface registration

In this section we discuss the problem of spatially aligning two digital surfaces with each other, referred to as 3D surface registration. Consider a moving reference model

$\mathcal{M}$ with vertices $M = [\boldsymbol{m}_1, \cdots, \boldsymbol{m}_{|M|}]^T$ and edges $\mathcal{E}$ which we want to transform to a fixed target model $\mathcal{F}$ with vertices $F = [\boldsymbol{f}_1, \cdots, \boldsymbol{f}_{|F|}]^T$.

1.3.3.1 Paired point matching

This section provides a way to align two point clouds with each other, that have the same number of points $N = |M| = |F|$ and that have known correspondences between them **?**. The point clouds centered around their center-of-mass are given by:

$$\tilde{\boldsymbol{m}}_i = \boldsymbol{m}_i - \frac{1}{N}\sum \boldsymbol{m}_i \tag{1.8}$$

$$\tilde{\boldsymbol{f}}_i = \boldsymbol{f}_i - \frac{1}{N}\sum \boldsymbol{f}_i \tag{1.9}$$

The 3×3 covariance matrix of the centered vertices $\tilde{M} = [\tilde{\boldsymbol{m}}_1, \cdots, \tilde{\boldsymbol{m}}_N]^T$ and $\tilde{F} = [\tilde{\boldsymbol{f}}_1, \cdots, \tilde{\boldsymbol{f}}_N]^T$ is given by: $S = \tilde{M}^T \tilde{F}$. Applying singular value decomposition (SVD) on the covariance matrix yields the following factorisation of S:

$$S = U \Sigma V^T, \tag{1.10}$$

with U and V being the left and right singular matrices. The diagonal matrix Σ contains the singular values. The optimal rotation and translation that brings the moving point cloud into alignment with the fixed point cloud is given by:

$$R = VU^T \tag{1.11}$$

$$\boldsymbol{t} = \frac{\sum \boldsymbol{f}_i}{N} - R\frac{\sum \boldsymbol{m}_i}{N} \tag{1.12}$$

1.3.3.2 Iterative closest point

Iterative closest point is an algorithm that can be applied when the correspondences between M and F are initially unknown. A rigid transformation that aligns both point clouds is estimated through the following iterative process:

1. First, the points of the source point cloud are matched to their closest neighboring points in the target point cloud. An efficient way to determine the closest points is by a Kd-Tree.
2. Next, the euclidean distance between the established pairs of corresponding points is being minimised. To this end, the rotation and translation parameters are estimated by the paired point matching procedure as discussed in section 1.3.3.1.
3. The moving points are updated by the above transformation.
4. The previous steps are repeated until the convergence criterion is met.

Many extensions to this basic formulation are proposed in the literature. One drawback of minimising the point-to-point distance is that it depends on the resolution of the two models, which can, for example, be avoided by minimising the point-to-plane distance instead.

1.3.3.3 Elastic surface registration

In this section we discuss an elastic registration method, proposed by Amberg et al **?**. In case of elastic registration, each vertex of the moving model has a translation vector $\boldsymbol{d}_i \in \mathbb{R}^3$ associated to it to model the deformation from the reference model towards the target. For each vertex $\boldsymbol{m}_i$ on the reference model, we look for the corresponding point $\hat{\boldsymbol{f}}_i$ on the target surface. As $\mathcal{F}$ is a mesh, this corresponding point does not necessarily need to be a vertex, but can lie on a triangular cell. One way to determine the points $\hat{F} = [\hat{\boldsymbol{f}}_1, \cdots, \hat{\boldsymbol{f}}_{|M|}]$ on $\mathcal{F}$ which correspond to the points M, is by casting a ray along each vertex normal and finding the intersection points with the target surface. Additional requirements on the intersection points can be included before considering them as corresponding points, such as requiring the same normal direction, and/or requiring no surface crossings.

After establishing a guess for the pairs of corresponding points, we minimise the registration energy function. The registration aims to minimise the distance between the moving and target surface:

$$E_d(D) = \sum_{m_i \in M} w_i ||\hat{\boldsymbol{f}}_i - (\boldsymbol{m}_i + \boldsymbol{d}_i)||^2 \tag{1.13}$$

with w_i equal to zero if vertex i has no corresponding point on $\mathcal{F}$, and equal to one otherwise. The deformation matrix $D = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_{|M|}]^T$ can be regularised by including an additional stiffness term, which penalises large deformations between neighboring vertices:

$$E_s(D) = \alpha \sum_{\{i,j\} \in \mathcal{E}} ||D_i - D_j||^2 \tag{1.14}$$

The previous equations can be combined in matrix notations as follows:

$$E[D] = \left\| \begin{bmatrix} \alpha G \\ W \end{bmatrix} D - \begin{bmatrix} 0 \\ W(\hat{F} - M) \end{bmatrix} \right\|_F^2 \tag{1.15}$$

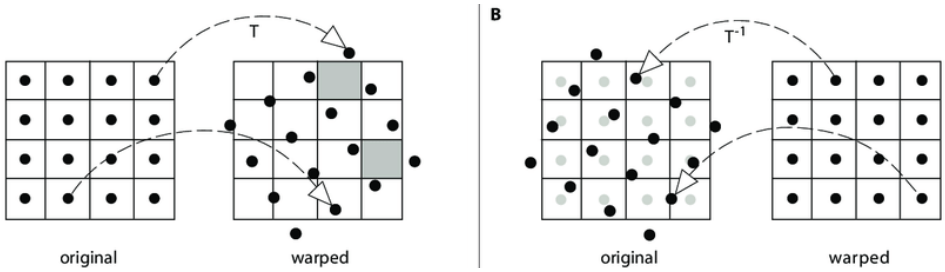
with $W = \text{diag}(w_1, \cdots, w_n)$ and where $||.||_F$ denotes the Frobenius matrix norm. The matrix G is the edge-connectivity matrix. If edge r connects vertices i and j, the matrix elements G_{ri} and G_{rj} equals 1 and -1 respectively. The other elements of row r are zero. The hyperparameter α balances the two energy functions with respect to each other, and changes during the iterative optimisation. In the beginning, the stiffness parameter α is large, and is gradually relaxed as soon as the surfaces come closer to each other. This allows to fine-tune the small scale deformations as soon as the larger structures are aligned.

1.3.4 Image registration

We now consider M and F to be a moving and fixed image, respectively, and try to find a transformation T that maps M to F.

1.3.4.1 Forward vs backward warping

In case of surface registration, the transformation T was applied on the individual points of the moving surface in order to match them with the target object. While the surface points are defined in the continuous space of $\mathbb{R}^3$, the pixel or voxel



(a) Forward warping. (b) Backward warping.

Figure 1.3: Difference between forward and backward warping.

coordinates in an image are not. They have discrete values and are defined on a discrete grid. As a result, applying the transformation T on an image coordinate $\boldsymbol{x}$, will not necessarily result in a position on the image grid anymore and, hence, holes can occur in the warped image, as illustrated in Figure 1.3. As a solution, backward warping of the moving image is the mainstream method in image registration. The moving image is sampled at locations, given by the inverse transformation of the fixed image coordinates. An interpolation scheme is used to interpolate the image intensity values at non-integer pixel positions in the moving image domain. This can be a linear interpolation, B-spline interpolation or nearest neighbor interpolation.

1.3.4.2 B-spline transformation

The local deformation of images can be described by displacement vector fields that relate voxels on the moving image to corresponding points in the fixed image domain. Free-form deformations control these local displacements through only a limited number of control points. This is realised by embedding the image in a coarse grid of control points. Changing one control point only affects the local neighborhood of that point. This leads to a sparse jacobian of the transformation and a more efficient calculation of the gradient of the registration metric.

B-spline transformations are a type of free-form deformations, where the voxel-deformations are a B-spline interpolation of the control point coefficients. Each control point on the coarse regular grid has a B-spline coefficient vector $\boldsymbol{C} \in \mathbb{R}^d$ associated to it, with d the number of image dimensions. The control point grid, with size $(L+3) \times (M+3) \times (N+3)$ and grid spacing $S_x \times S_y \times S_z$, partitions the original image into tiles such that each tile comprises multiple voxels. The deformation of any voxel in a tile is determined by the $(n+1)^d$ closest control points, with n being the order of the spline. Cubic splines ($n = 3$) have 16 and 64 control points per voxel for the 2D and 3D case, respectively. A 1D cubic spline is defined as a piece-wise polynomial of the basis-functions B_n:

$$B_0(u) = (1-u)^3/6 \tag{1.16}$$

$$B_1(u) = (3u^3 - 6u^2 + 4)/6 \tag{1.17}$$

$$B_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6 \tag{1.18}$$

$$B_3(u) = u^3/6. \tag{1.19}$$

with $u = \frac{x}{S_x} - \lfloor \frac{x}{S_x} \rfloor \in [0, 1[$ being the relative position of (x, y, z) within a tile of the

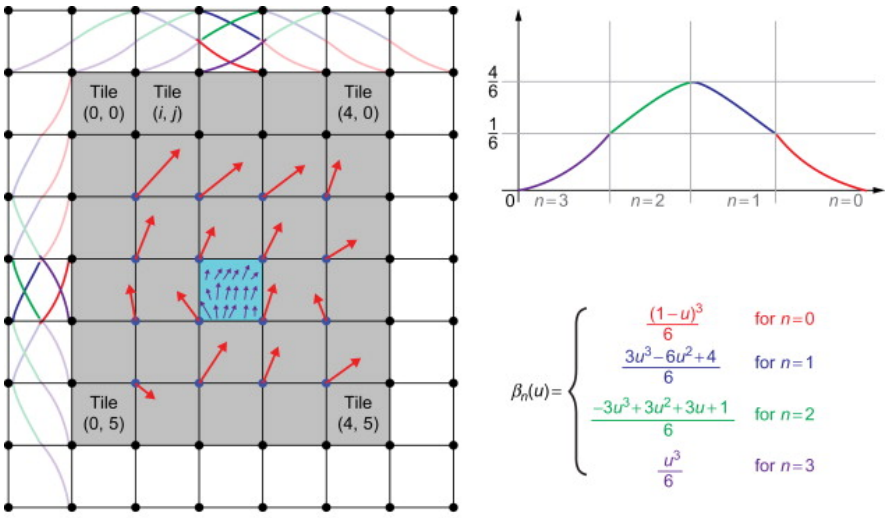


Figure 1.4: B-spline based free form deformation of an image **?**.

grid. The same formulas apply to the basis functions in the y and z directions. The deformation at position (x, y, z) is given by the 3D tensor product of those 1D cubic B-splines:

$$d_j = \sum_{r=0}^{3}\sum_{s=0}^{3}\sum_{t=0}^{3} B_r(u)B_s(v)B_t(w)C_{l+r,m+s,n+t} \tag{1.20}$$

with $l = \lfloor \frac{x}{S_x} - 1 \rfloor$, $m = \lfloor \frac{y}{S_y} - 1 \rfloor$ and $n = \lfloor \frac{z}{S_z} - 1 \rfloor$ being the grid control point indexes surrounding image position (x, y, z).

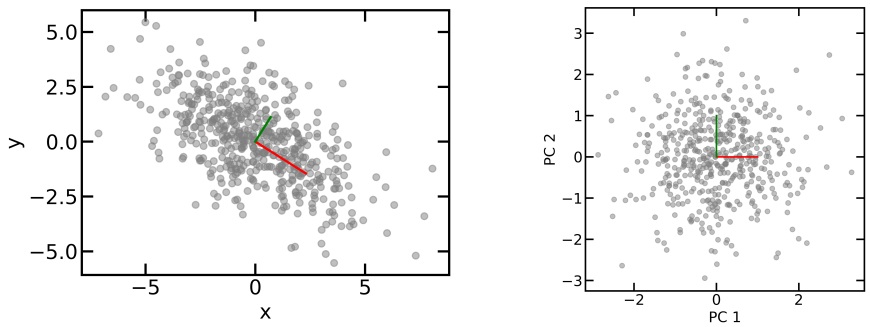
1.4 Shape statistics

Statistical shape modeling is a powerful tool to understand the shape variability in a population of subjects **?**. Instead of describing the spatial variation of each single vertex of a shape, it looks for common variation modes across all vertices, resulting in only a limited number of parameters to describe the shape variation. To understand this type of statistics, we first give a general introduction to principal component analysis, before applying it to shape modeling.

1.4.1 Principal component analysis

Principal component analysis (PCA) is a statistical method to reduce the dimensionality of a dataset with correlated variables. By applying an orthogonal transformation, it transforms the data into a smaller set of linearly-uncorrelated variables. Assume a data matrix $X \in \mathbb{R}^{M\times N}$, with M the number of variables and N the number of samples. We assume that the rows of X are zero-centered, meaning that the row means are subtracted from the data. In that case, the covariance matrix $C \in \mathbb{R}^{M\times M}$ can be written as:

$$C = \frac{XX^T}{N-1} \tag{1.21}$$



(a) Original, correlated (x, y)-data. (b) Coordinate transformation to PC-space.

Figure 1.5: Example of principal component analysis on two correlated variables x and y. The principal axes/eigenvectors, shown in red and green, are scaled by their corresponding standard deviations $\sqrt{\lambda_i}$.

As the covariance matrix is symmetrical, it can be diagonalised as follows:

$$C = ULU^T \tag{1.22}$$

where L is a diagonal matrix, containing the eigenvalues λ_i. The columns of U correspond to the eigenvectors, which are called the principal axes. The projections of the data X onto those principal axes are the principal component scores and are given by:

$$\tilde{X} = X^T U \tag{1.23}$$

The columns of $\tilde{X}$ are the principal components. The i^{th} row gives the coordinates of the i^{th} datapoint in the PC space. This new coordinate system expresses the data with respect to uncorrelated variables.

The principal components are commonly calculated by SVD on X, which is given by: $X = USV^T$, where U and V are the left and right singular vectors and S is a diagonal matrix containing the singular values s_i. Given the SVD, the covariance matrix can be rewritten as:

$$C = \frac{XX^T}{N-1} = \frac{1}{N-1} USV^T VSU^T = U\frac{S^2}{N-1}U^T, \tag{1.24}$$

where we used the fact that V is an orthogonal matrix. Comparing with eq.(1.22) we can identify that the left singular vectors of the SVD must correspond to the principal directions. Furthermore, the singular values are correlated to the eigenvalues of the covariance matrix as follows:

$$\lambda_i = \frac{s_i^2}{N-1} \tag{1.25}$$

The eigenvalues of the covariance matrix thus show the variances of the respective principal components. The principal component scores of eq.(1.23) can also be expressed in terms of the right singular vectors:

$$\tilde{X} = X^T U = VSU^T U = VS \tag{1.26}$$

The principal components are ordered with decreasing variance, meaning that the first principal component accounts for the maximum amount of variance in the dataset.

Each succeeding principal component describes the maximum amount of variance under the constrained that it is orthogonal to all preceding principal components. As a result, later principal components are less significant and often represent the noise in the dataset. This allows expressing the data by only the first k PCs with $k < M$. We therefore select the first k columns of V and the $k \times k$ upper-left part of S.

Often, the eigenvectors or principal axes are scaled by their respective standard deviation $\sqrt{\lambda_i} = s_i/\sqrt{N-1}$, such that the different principal components have the same range. The standardised PC scores with respect to the normalised principal axes are given by $\sqrt{N-1}V$, which follows from eq.(1.26). An example of normalised principal axes for two correlated variables is shown in Figure 1.5.

While being the most frequently used method for dimensionality reduction, PCA is not the only option. Kernel principal component analysis (KPCA) for example, can be applied on more complex clustered data that can not be properly transformed into a linear subspace, spanned by the usual principal components. Dimensionality reduction on non-gaussian-distributed data can be achieved by independent component analysis (ICA).

1.4.2 Statistical shape models

Statistical shape models (SSM) describe the shape variations in a population by means of variation modes. The calculation of those modes involves two steps on a training set of example shapes. First, the training shapes must be registered to each other by any method as outlined in section 1.3.3, such that each subject is described by the same semantic-meaningful set of points. Secondly, a dimensionality-reduction method, outlined in section 1.4.1, must be applied on the set of registered shapes. In case of PCA, the matrix $X \in \mathbb{R}^{3M \times N}$ is now composed of the linearized vertex coordinates, with M the number of vertices and N the number of training subjects. After deriving the principal components, any new shape can be expressed as the sum of the mean shape and a linear combination of the variation modes:

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \sum_{i=1}^{N} \alpha_i \sqrt{\lambda_i} \boldsymbol{u}_i \tag{1.27}$$

with λ_i and $\boldsymbol{u}_i$ being the normalised eigenvalues and eigenvectors of the PCA, respectively.

While SSMs were originally built on only a sparse set of landmarks, they can also be built on denser set of surface points. Besides the statistical shape information, the models can also include statistical appearance information. Such "statistical shape and appearance models" (SSAM) are built on an observation matrix X containing vertex coordinates and appearances, like the Hounsfield units or bone mineral density for example. The statistical modeling method as outlined here also holds in the image domain. In this case the matrix X contains the B-spline coefficients of the deformation fields that align all subjects with each other. The resulting model is called a "statistical deformation model" (SDM) **?**.

Statistical shape models are useful for morphometric studies, where they provide more insights in the geometric variations compared to extracted measures as length, radius, etc. From the statistical model, one can generate an arbitrarily large database of virtual shape instances, which can help, for example, in determining to which

portion of a population a certain implant best fits **?**. These artifical databases are in particularly interesting to train deep-learning models on (see chapter 3).

While statistical shape models are usually built for bony structures, they can be extended with cartilage and ligament information to obtain a complete musculoskeletal model for biomechanical simulations **?**. The correlation between the joint morphology and its biomechanical function can help in understanding why some people have more risk for a certain injury or pathology, and can help in improving the diagnosis and treatment process **?**.

Besides explaining the statistical shape variation in the population, a SSM can also act as a prior model in the reconstruction of a person-specific model. This is a very efficient type of registration, thanks to the limited number of shape parameters in a SSM. In this case, we speak of "active shape model" (ASM) or "Active appearance models" (AAM). The latter helps the registration process by exploiting the appearance information. Similar to the atlas-based segmentation, discussed in chapter 1.3.1, this allows for automatic segmentation and pre-operative planning **?**.

The shape information of these statistical models can also be combined with finite element models to personalise mechanical simulations. Finite element models are used to simulate physical processes, like for example the cartilage stress and strain in a joint, by solving the appropriate differential equations on the discretised shape. Finite element models are in general time-consuming to construct, but can be modified according to the statistical shape information. In the biomedical context, this is in particular interesting for osteoarthritis studies, for example, to study the relation between bone shape and stresses or to assess the person-specific risk for bone fractures **??**.

2

X-ray imaging

Contents

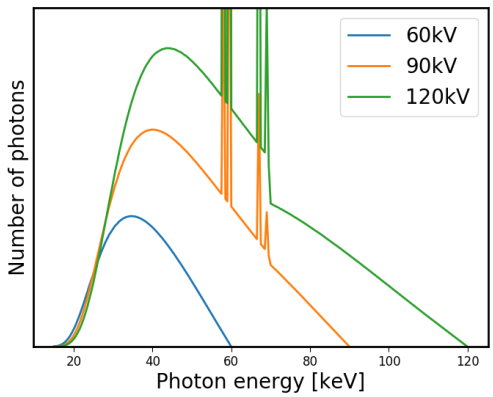


Figure 2.1: X-ray spectra for different generator voltages. Data from **?**.

2.1 X-ray physics

X-rays are a type of electromagnetic radiation, widely used for biomedical imaging because of its ability to visualise the internal parts of a patient. Compared to visible light, X-rays have a higher frequency, and therefore a higher energy. In this section we discuss the production of X-rays, their interaction with matter and how they can be detected for the purpose of medical imaging.

2.1.1 X-ray production

An X-ray generator circuit consists of two electrodes: a negative cathode and a positive anode. The cathode filament is heated up by an electric current to temperatures around 2000°C. The high kinetic energy of the electrons at the cathode surface enables them to escape from the cathode, which is known as thermionic emission. The freed electrons are subsequently accelerated towards the positive anode through a voltage of several kilo-volts, and thereby gain an energy equal to the voltage times the electron charge:

$$E_e = e \cdot V = V \frac{\mathrm{eV}}{\mathrm{V}}, \tag{2.1}$$

where the unit electronvolt (eV) is defined as the energy given to a fundamental charge accelerated through a potential difference of 1 V. The high-energy electrons bombard the anode, which is made of a material with high-atomic number and high melting temperature, like tungsten or molybdenum. The interaction of high-energetic electrons with the anode material leads to the production of X-rays in a certain energy spectrum which depends on the applied voltage between the anode and cathode. The spectrum, shown in Figure 2.1, consists of a broad continuous spectrum and single characteristic lines.

The bulk spectrum is caused by Bremsstrahlung, a process in which the electrons are decelerated by the Coulomb field of the positive protons. The deceleration means a loss of kinetic energy, which is compensated by the emission of X-ray photons. Because of energy conservation, the maximal photon energy equals the electron energy, which

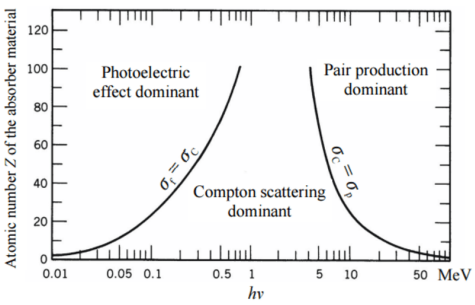


Figure 2.2: Relative importance of the 3 main interaction processes of X-rays and gamma-rays. At the boundary lines between two processes the cross-sections are equal. Extracted from **?**.

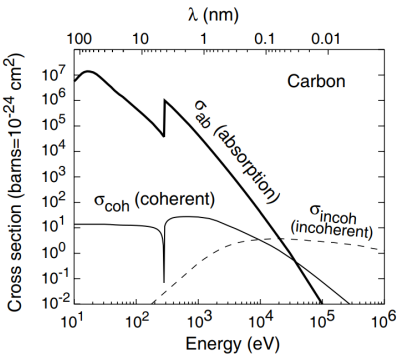


Figure 2.3: X-ray interaction cross-sections in carbon **?**.

is equal to Ve. Most likely, this total energy is divided over multiple photons being emitted which therefore must have a lower energy.

The sharp peaks in the spectrum are associated to characteristic X-ray emission. When the energy of the incoming electrons is higher than the binding energy of the inner-shell electrons of the anode material, the inner-shell electrons can get ejected. In that case, it leaves a hole which can be occupied by a higher-shell electron. By occupying a lower-energy shell an X-ray photon will be emitted with an energy equal to the energy difference between those two shells. As the energy levels of the atoms are fixed, the emitted photons can only have these characteristic energies.

2.1.2 Interaction of X-rays with matter

While traveling through matter, X-rays undergo various sorts of interactions with the material, resulting in an attenuation or reduction of the beam intensity. The dominant interaction process depends on the atomic number of the material and the X-ray energy, as can be read from Figure 2.2.

- The photoelectric effect is the dominant interaction process at the low energy side of the X-ray spectrum, where also medical X-rays belong to. In a photo-electric interaction, the X-ray photon kicks out an inner-shell electron from the atom, while the photon itself gets completely absorbed during this process. Part of its energy is used to overcome the binding energy of the electron to the atom and the remaining part is converted into kinetic energy of the ejected electron. For a typical medical X-ray beam energy of 40 keV, the outer layer of bones (i.e. cortical bone) with an effective atomic number of 10.4 will undergo more photo-electric interactions than soft tissue with an effective atomic number of 4.7 **?**.
- Coherent scattering: When the energy of the X-ray photons is low compared to the binding energy of the outer-shell electrons, there is not enough energy to

eject the electron. The photon changes direction but preserves its energy and momentum.

- Incoherent scattering / Compton scattering: In Compton scattering the photon loses part of its energy to an outer shell electron, which is only loosely bound to the atom. The electron is kicked out of the atom and the photon gets deflected. The energy lost by the photon depends on the scatter angle. The more energetic the photon, the more forward the scattering will be. The photon loses the least amount of energy when the scatter angle is small and is maximal when it changes 180 degrees in direction. Compton scattering is a common interaction, which occurs at all energies in all materials.
- Pair production: In pair production, the photon interacts with the electric field of the atom. The photon vanishes and creates an electron/positron pair. It is most likely to happen at high photon energy and for high atomic number materials.

2.1.3 X-ray detection in radiology

Radiograph images, also called RX or X-ray images, capture the remaining X-ray photons after having passed through a patient. The dependency of the X-ray attenuation on the type of material makes it possible to differentiate between organs, bones, etc. on such image. For the first 90 years following the discovery of X-rays in 1895, the X-ray detection was based on chemical processes in a light-sensitive emulsion **?**. While initially on glass photographic plates, the substrate for the emulsion was later replaced by a photographic film. In 1970s, computed radiography was developed which could digitize the attenuation pattern stored in the phosphor image plate through laser stimulation. A digital representation of the attenuation pattern made it possible to more easily share and store information, but still required manual replacement of the plates in between acquisitions.

In 1990s, with the advancement in the semiconductor technology, flat-panel detectors were developed, starting the digital radiography era. A flat panel detector converts the X-ray photons into electric charges which are read out by a thin-film transistor array.

An alternative to flat panel detectors include X-ray image intensifiers, which convert the X-rays into visible light and amplify the intensity to a measurable magnitude. These are mostly used in fluoroscopy acquisitions, which, in contrast to radiography, image the object in motion. The amplification of the signal is realised by a photomultiplier tube (PMT). It consists of a negatively charged photocathode with a phosphor coating which converts the photon in a photo-electron, and a series of positively charged dynodes, which causes an avalanche of secondary electrons for each incident electron.

It is important to note that radiographs or fluoroscopy images represent projections of the patient on a 2D plane, hence, many anatomical structures overlap with each other. All the 3D information gets piled up along the beam axis. However, by acquiring many 2D projections at different angles around the patient, the 3D attenuation profile can be reconstructed through computed tomography (CT)-reconstruction.

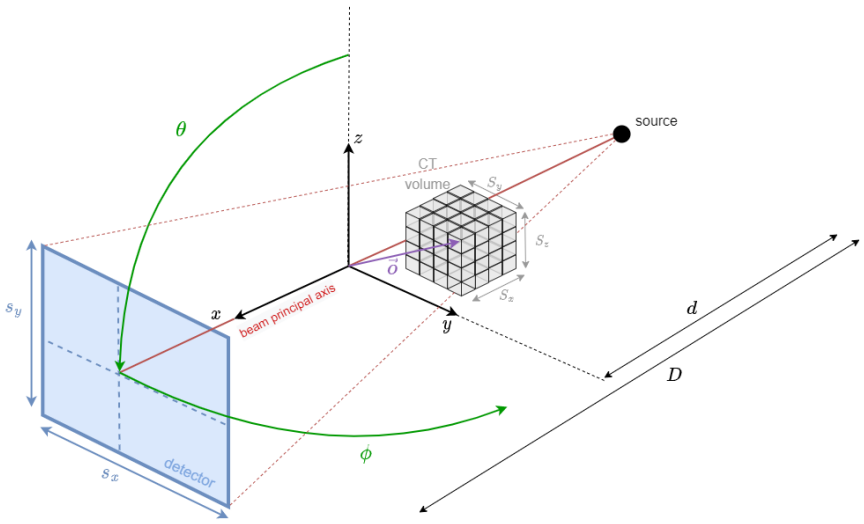


Figure 2.4: Conebeam projection geometry.

2.2 Radiographic projection

To image a patient by X-rays, the patient is positioned between an X-ray source and a detector. The X-ray beam can either consist of parallel or diverging X-rays. Figure 2.4 illustrates a cone-beam projection geometry in which the X-rays originate from a point source and diverge towards the detector plane. In medical set-ups, the source-detector system is mounted on a gantry in order to easily acquire radiographs from different directions around the patient.

In Figure 2.4, the world coordinate system is centered at the isocenter, being the intersection point of the beam principal axis and the gantry rotation axis. The source and detector are aligned such that the beam principal axis is perpendicular to the detector plane and intersects the plane at its center.

The cone-beam projection of the patient onto the 2D detector plane is mathematically described by a perspective projection camera matrix P, that maps every 3D world coordinate to a 2D image coordinate. This involves transformations between three coordinate systems: the world, camera and image reference frame. The projection matrix P can be decomposed into the two different coordinate transformations: the extrinsic transformation R that describes the position and orientation of the source-detector system in the world reference frame, and the intrinsic transformation K that describes how the image is captured, in terms of focal length, sensor resolution, etc.

2.2.1 Extrinsic parameters

The extrinsic transformation seeks a transformation R that brings the data from the world coordinate frame to a canonical camera reference frame, in which the source is positioned at the origin and the beam principal axis is aligned with the z-axis. This transformation can be parameterised by two spherical angles: left/right anterior oblique (LAO/RAO) angle ϕ and cranial/caudal (CRAN/CAUD) angle θ. They

describe the orientation of the gantry with respect to the patient.

The transformation of the world reference frame to the canonical camera reference frame consists of a rotation by angle θ around the axis $[\sin(\phi), \cos(\phi), 0]$, followed by a rotation by angle ϕ around the beam principal axis:

$$R = R_z \begin{bmatrix} \sin^2\phi(1-\cos\theta)+\cos\theta & -\sin\phi\cos\phi(1-\cos\theta) & -\cos\phi\sin\theta \\ -\sin\phi\cos\phi(1-\cos\theta) & \cos^2\phi(1-\cos\theta)+\cos\theta & -\sin\phi\sin\theta \\ \cos\phi\sin\theta & \sin\phi\sin\theta & \cos\theta \end{bmatrix} \tag{2.2}$$

with:

$$R_z = \begin{bmatrix} \cos(-\phi+\pi/2) & -\sin(-\phi+\pi/2) & 0 \\ \sin(-\phi+\pi/2) & \cos(-\phi+\pi/2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

The translation part of the extrinsic transformation accounts for the offset $\boldsymbol{o}$ between the patient and the isocenter, defined in the camera reference frame, and brings the source to the origin of the camera reference frame:

$$\boldsymbol{t} = \begin{bmatrix} o_x \\ o_y \\ o_z + d \end{bmatrix}, \tag{2.4}$$

with d the distance between the source and isocenter.

2.2.2 Intrinsic parameters

The cone-beam projection in the canonical frame can be identified as a pinhole-camera with the source-detector distance D as focal length. In the 2D case, Thales's theorem proves that the projection of a point (x, z) along the z-axis onto a plane at a distance D from the source, is given by Dz/y. The general solution can be written as a 3×4 matrix multiplication of homogeneous coordinates:

$$K = \begin{bmatrix} 1/\delta_x & 0 & s_x/2 \\ 0 & 1/\delta_y & s_y/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D & 0 & 0 & 0 \\ 0 & D & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.5}$$

The first part of this camera intrinsic matrix transforms the physical coordinates to unit-less pixel coordinates and aligns the center of the image plane with the optical center. The image plane sensor is characterised by sensor size (s_x, s_y) and pixel size (δ_x, δ_y).

2.2.3 Projection matrix

The extrinsic and intrinsic transformations together define the cone-beam projection, that transforms a world coordinate to the image plane:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} D/\delta_x & 0 & s_x/2 & 0 \\ 0 & D/\delta_y & s_y/2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.6}$$

Note that the result is a homogeneous coordinate (u, v, w). Conversion to Euclidean coordinates involves division of the coordinates by the third coordinate, yielding the image coordinates: $(u/w, v/w)$.

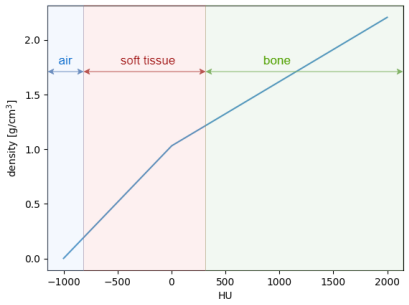


Figure 2.5: Conversion relation between Hounsfield units (HU) and density **?**.

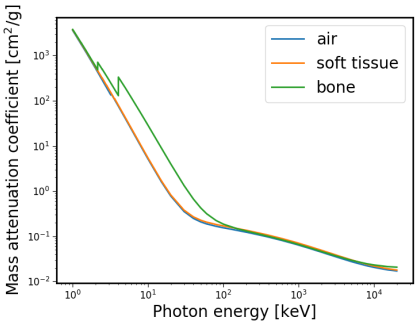


Figure 2.6: Energy-dependent mass-attenuation for different material types **?**.

2.3 Radiograph simulation

The development and validation of new computer methods in radiology often requires a large number of radiographs with ground-truth labeling. Manually annotating such dataset can be time-consuming. Secondly, a projection image with overlapping structures can be difficult to interpret, resulting in subjective labeling. Instead of acquiring radiographs, researchers therefore often rely on synthetic radiographs generated from a surface model or 3D image, generally called "digitally reconstructed radiographs" (DRR). In this section we describe a ray-casting method to generate DRRs **?** and conclude with limitations of this method.

2.3.1 Ray casting

DRRs can be simulated as perspective projections of a 3D image onto the 2D image plane. Ray casting simulation methods shoot X-rays as straight lines from the source to the detector pixels, while the intensity along each ray gets attenuated due to the various interaction mechanisms between photons and matter. The decrease in X-ray intensity can be understood as follows. After each infinitesimal step dx, a fraction of the number of X-ray photons is lost. The loss of beam intensity is proportional to the particle number density n of the medium, the beam intensity I and the path length dx:

$$dI \propto -nIdx. \tag{2.7}$$

The proportionality constant reflects the probability of a photon being scattered or absorbed, and is often combined with the particle number density n into the linear attenuation coefficient μ. This material-specific constant is a measure for how easily X-rays can penetrate through the material. Integrating eq.(2.7) over the path length x of the ray yields:

$$\int_{I_0}^{I} \frac{dI}{I} = -\int_{0}^{L} \mu(x)dx, \tag{2.8}$$

where I_0 is the initial beam intensity at the source location. This results in the following expression for the beam intensity I at a radial distance L from the source:

$$I = I_0 e^{-\int_0^L \mu(x)dx}. \tag{2.9}$$

This expression describes an exponential decrease in X-ray intensity with the distance travelled through the medium. In case of a homogeneous object, i.e. having a uniform density and atomic number, this reduces to the well-known Lambert-Beer law:

$$I = I_0 e^{-\mu L}. \tag{2.10}$$

As the attenuation coefficient μ is zero in vacuum, the total path length L in this equation is equal to the Euclidean distance between the points where the ray enters and leaves the object. This equation can be used to simulate radiographs from surface models which describe objects by their outer hull without density information.

In contrast to surface models, volumetric models allow to simulate radiographs of non-homogeneous objects. In models composed of tetrahedral cells, for example, the attenuation profile per tetrahedron can be described by a linear combination of Bernstein basis polynomials $B^d_{\boldsymbol{k}}$ (see section 1.2.1). The integral from eq.(2.9) for a single tetrahedron can then be evaluated as follows:

$$\int_{\boldsymbol{p_0}}^{\boldsymbol{p_1}} \mu(\boldsymbol{u}) d\boldsymbol{u} = \sum_{|\boldsymbol{k}|=d} \beta_{\boldsymbol{k}} \int_{\boldsymbol{p_0}}^{\boldsymbol{p_1}} B^d_{\boldsymbol{k}}(\boldsymbol{u}) d\boldsymbol{u} \tag{2.11}$$

$$= \sum_{|\boldsymbol{k}|=d} \beta_{\boldsymbol{k}} \frac{||\boldsymbol{p_1} - \boldsymbol{p_0}||}{d+1} \sum_{\boldsymbol{l} \subseteq \boldsymbol{k}} B^{|\boldsymbol{l}|}_{\boldsymbol{l}}(\boldsymbol{u_0}) B^{|\boldsymbol{k}-\boldsymbol{l}|}_{\boldsymbol{k}-\boldsymbol{l}}(\boldsymbol{u_1}) \tag{2.12}$$

with $(\boldsymbol{u_0}, \boldsymbol{u_1})$ being the barycentric coordinates of the intersection points $(\boldsymbol{p_0}, \boldsymbol{p_1})$.

Volumetric images, on the other hand, are discretised into 3D volume elements, called voxels. In a computed tomography (CT) image, the radiodensity of each voxel is expressed by its Hounsfield unit (HU), which is a linear transformation of the linear attenuation coefficient, such that, by definition, -1000 HU and 0 HU correspond to radiodensity of air and water, respectively. The integral of eq.(2.9) is taken over all volume elements intersected by the ray:

$$\int_{p_0}^{p_1} \mu(\boldsymbol{u}) d\boldsymbol{u} = \sum \mu_i dx_i \tag{2.13}$$

with dx_i being the pathlength of the ray through volume element i. The linear attenuation coefficient μ_i is equal to the mass attenuation coefficient μ_m multiplied by the mass density ρ. The mass density can be calculated from the Hounsfield units of the CT image by the conversion graph of Figure 2.5. The mass attenuation coefficient μ_m is energy- and material-dependent and is shown in Figure 2.6. Notice the similar features with the graph of Figure 2.3, showing the interaction cross-sections. As the mass attenuation coefficient depends on the type of material, the computation of eq.(2.13) requires a labeling of the different voxels according to their material types. This so-called segmentation of the CT data can be obtained by simply applying different thresholds on the HU values as indicated in Figure 2.5:

$$M(x) = \begin{cases} \text{air}, & \text{if } HU(x) \leq -800, \\ \text{soft tissue}, & \text{if } -800 < HU(x) \leq 350, \\ \text{bone}, & \text{if } HU(x) > 350. \end{cases} \tag{2.14}$$

In case the object consists of different types of material, the integral of eq.(2.9) must be evaluated for each material separately and multiplied by the mass attenuation

(a) air (b) soft tissue (c) bone

Figure 2.7: Integrated density along the rays for each class of materials.

coefficient:

$$I = I_0 \exp\Big(- \sum_{m \in M} \mu_m(E) \int_0^L \delta(m, M(x)) \rho(x) dx \Big). \tag{2.15}$$

Each term of the sum calculates the attenuation due to a different material m, being either bone, soft tissue, or air. Examples of the density integrals for the different materials are shown in Figure 2.7.

A polychromatic X-ray beam is composed of different energies, as shown in Figure 2.1. As the attenuation coefficient is also dependent on the energy, we need to evaluate the total attenuation of eq.(2.15) for every energy bin of the spectrum. Integrated over the entire energy spectrum yields:

$$I = \int p_0(E) \exp\Big(- \sum_{m \in M} \mu_m(E) \int_0^L \delta(m, M(x)) \rho(x) dx \Big) dE, \tag{2.16}$$

with $p(E)$ the spectral density. Note that the path integral must only be calculated once per material class, as it became energy-independent after introducing the mass-attenuation coefficient.

2.3.2 Secondary effects

There are some limitations to the simulation framework as outlined above, due to other physical processes which would contribute to a real radiograph. As seen in section 2.3.1, ray-casting methods model the photon-matter interactions as an effective attenuation along a straight line. However, as seen in section 2.1.2, X-rays can undergo a change in direction through coherent or incoherent scattering. This can only be modeled by probabilistic methods, like Monte Carlo (MC) methods, which simulate the trajectory of every single photon and evaluate the probability of scattering for each photon-matter interaction. To avoid time-consuming probabilistic simulations, recent deep-learning methods have been developed which, trained on MC outputs, can estimate an effective scatter image **?**.

Another limitation is the effect of "beam hardening". As low-energy photons are more likely to get attenuated than the high-energy photons (cfr. Figure 2.6), it is expected that at large penetration depths the ray will mostly consist of higher energetic photons. This phenomenon makes the spectral density $p(E)$ in eq.(2.16) dependent on the path-length x.

Finally, real radiographs are also subject to quantum noise and electronic read-out noise.

3

Deep Neural Networks

Contents

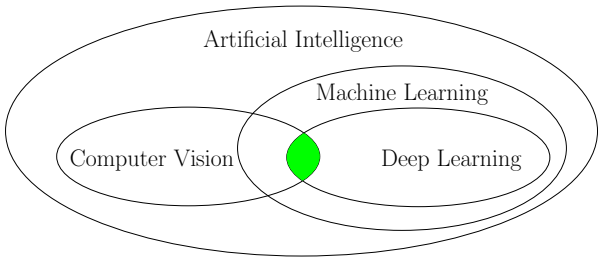


Figure 3.1: Venn diagram with the different subfields of artificial intelligence. The green region indicates the subfield that this thesis focuses on.

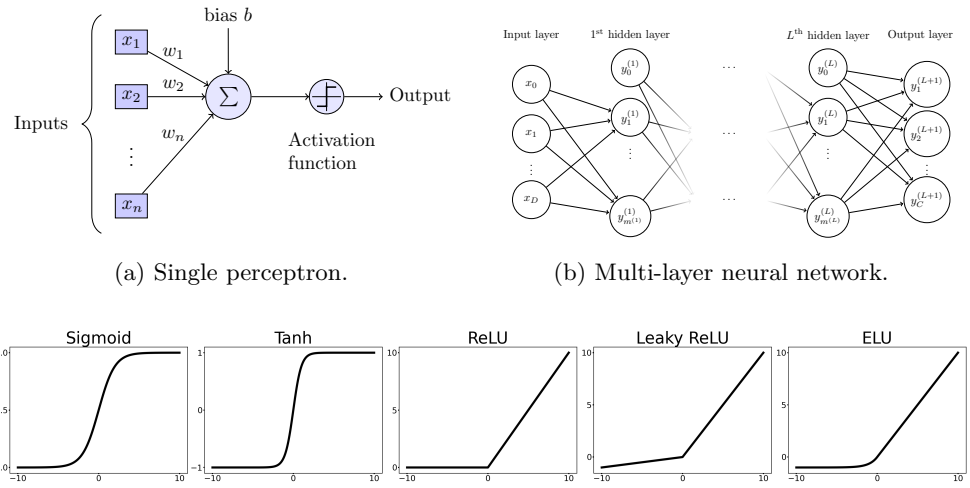
3.1 Introduction

Artificial intelligence (AI) is a field within computer science research concerning systems which perceive their environment and take appropriate actions autonomously to achieve their goal. AI was founded as such during the Dartmouth summer conference in 1956 **?**. At this event 11 mathematicians and scientists came together in an attempt to let computers mimic the human decision taking process. In the following years different processes were taken over by computers, which could only be done by humans until then. Natural language processing (NLP) and speech recognition were one of the first applications of AI. The first mobile AI-based robot saw the light of day in 1972 and listened to the name "Shakey". It was able to perceive its environment and accomplish tasks without step-by-step instructions. Despite the important AI milestones, AI research started to decline due to a lack of funding, which is known as the AI winter.

In 1990s, with the development of probability theory and statistics, a new type of AI arose, namely machine learning. The goal of ML is to learn underlying processes from a large amount of example data. However, many early ML algorithms were not strong enough to learn directly from the data and needed manual extraction of features. These are details in the input data that are believed to be important to understand the underlying process. ML models use decision trees, support-vector machines (SVM), neural networks (NN), etc to process these features to come up with a solution. A typical workflow consists of training, validation and testing.

The growing availability of large labeled databases and the technological advances in computing power through CPU's and GPU's, made it possible to make neural networks deeper and more complex with multiple layers. The benefit of this new type of ML, called Deep learning, is that it does not require manual feature selection anymore. Instead, the network is deep and complex enough to learn the relevant features itself from the input data. DL gained popularity over classical methods as they achieve higher accuracy and can outperform human decision making. More so than other methods, the DL's accuracy benefits from the large availability of data.

Deep-learning became a successful technique in many computer vision problems. While DL teaches machines to learn from data experiences, computer vision teaches machines to understand visual data like images and videos. Types of problems that DL can be applied to in the field of computer vision includes for example image segmentation,



(a) Single perceptron.

(b) Multi-layer neural network.

(c) Activation function types.

Figure 3.2: Building blocks of a neural network. (a) A single perceptron calculates the weighted average of its input and applies an activation function. (b) Perceptrons can be stacked into multiples layers in order to solve more complex problems. (c) Different types of non-linear activation functions.

registration, object detection, landmark detection and classification.

3.2 Artificial neural networks

Similar to how our brain processes information through a network of biological neurons, an artificial neural network solves a complex task by feeding input data to several layers of artificial neurons. Each single artificial neuron or perceptron, shown in Figure 3.2a, computes a weighted sum of its inputs:

$$y = b + \sum w_i x_i \tag{3.1}$$

The weights w_i and offset b are internal parameters of the perceptron and express the importance of each input node to the output. The output is passed to a nonlinear activation function, which allows the network to solve more complex tasks. Without the non-linearity, the network would behave as a linear model irrespective of the number of layers. Several activation functions have been proposed in the history of neural network research, including sigmoid-function, tanh-function, rectified linear unit (ReLU), Leaky-ReLU and exponential linear unit (ELU) **?**. Their behavior is shown in Figure 3.2c.

In order to solve more complex tasks, artificial neural networks (ANN) stack multiple layers of perceptrons behind each other, as illustrated in Figure 3.2b. The universal approximation property of neural networks states that a two-layer perceptron with only one hidden layer is already capable to approximate any continuous function to any desired accuracy **?**. Each perceptron of a layer in an ANN is connected with each perceptron of the previous layer, which is the reason why those layers are called “fully

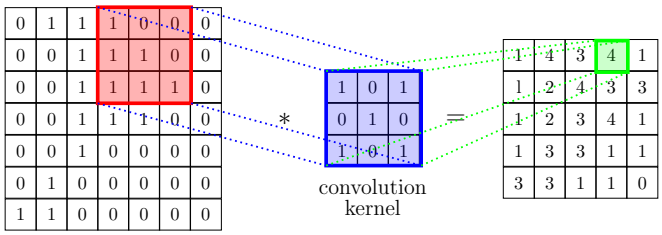


Figure 3.3: Convolutional operation between an input (left) and a kernel (blue).

connected layer". Each connection has a certain weight associated which needs to be learned during the training process.

3.3 Convolutional neural networks

Convolutional neural networks (CNN) are a specific type of network architure that has been optimised for images. They exploit the spatial relationship between pixels in 2D images or between voxels in 3D image volumes. In this section we discuss the main building blocks of a CNN.

3.3.1 Convolutional layer

A convolutional layer extracts multi-scale localised spatial features from the input data, taking into account neighboring pixels or voxels. It does this by applying a discrete convolution between the input feature image g and a small window kernel f, visualised in Figure 3.3:

$$(f * g)[x, y] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} f[n, m] \cdot g[x - n, y - m] \tag{3.2}$$

$$(f * g)[x, y, z] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[n, m, l] \cdot g[x - n, y - m, z - l] \tag{3.3}$$

Note that, in contrast to fully-connected layers in an ANN, a perceptron is only connected to a small subset of perceptrons of the previous feature map. The kernel g determines what feature the layer will filter from its input. The layer will produce a strong response for locations in the input where it finds this patch. The weights of the layer's kernel are regarded as the internal parameters of the layer and need to be optimised during training of the network. Training of the network will learn which kernels are most useful for the specific problem. Besides, the convolutional layer has also some non-learnable hyperparameters that control its behavior:

- The **kernel size** K determines the size of the sliding window.
- The **stride** s determines how many pixels the kernel window is shifted after each convolution step. This is normally equal to one, such that no information is lost.
- The **dilation** d determines the spacing between kernel elements.

- **Padding** P of the input image will add zeros to the borders of an image, such that no information is missed when sliding the kernel window over the image.
- The desired **number of output features** F_{out} is the number of filters and controls the number of kernels the convolutional layer will have, or in other words: how many patterns the layer will filter from the input.

The size of the output feature map of a convolutional layer also depends on these hyperparameters. The size of the output feature map is given by $(H_{out}, W_{out}, F_{out})$, with:

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times P[0] - d[0] \times (K[0] - 1) - 1}{s[0]} + 1 \right\rfloor \tag{3.4}$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times P[1] - d[1] \times (K[1] - 1) - 1}{s[1]} + 1 \right\rfloor \tag{3.5}$$

The number of parameters used by the convolutional layer, including biases, equals to:

$$n = (K^2 F_{in} + 1) F_{out}. \tag{3.6}$$

Stacking two 3×3 convolutions behind each other has the same receptive field as a single 7×7 convolution, meaning that the area of the first feature map seen by a single pixel in the last one has equal size. However, according to the previous formula the 3×3 convolutions have less weights to be learned than the single 7×7 convolution. Secondly, the 3×3 convolutions will be better in learning complex concepts which increases the learning capacity, as a non-linear activation can be included after each convolution. It is therefore preferred to cascade multiple small convolutions instead of one large convolution.

3.3.2 Pooling layer

Pooling layers reduce the spatial dimensions of the feature maps without changing the number of features. Like a convolutional layer, it slides a window across the feature map, but now filters the values within that window by only keeping the maximum value (max-pool layer) or their average value (average-pool layer). Note that this operation does not involve learnable parameters, which makes it an easy way to reduce spatial information and save computational cost. The downside of this compressing operation however is the loss of information.

3.3.3 Batch normalisation

In general, it is a good practice to normalise the input data to a network in order to reduce the complexity of the data distribution and to simplify the decision boundary that the network is supposed to model. The different scales of the input data can otherwise result in slow and unstable training. The same reasoning applies to the intermediate feature maps in the network. The normalisation of those feature maps are taken care of by the batch normalisation layers. They ensure a more efficient training which allows for larger training rates and which is less sensitive to its parameter initialisation **?**. The batch normalisation was originally proposed to be added right before the activation function.

The batch normalisation layer normalises each channel of its input tensor $I_{b,c,x,y}$, by subtracting its mean $\mu_c = \frac{1}{N}\sum_{b,x,y} I_{b,c,x,y}$ and deviding the zero-centered data by the standard deviation σ_c. This results in all channels to have a similar range.

After the normalisation, it applies a linear transformation on each channel, parameterised by the hyper-parameters γ_c and β_c. These learnable parameters are optimised during training and allow to control the activation distribution. It prevents, for example, the data distribution to fall in the linear regime of a ReLU activation function.

Putting the two parts, together, the output of the batch normalisation layer can be written as follows:

$$O_{b,c,x,y} = \gamma_c \frac{I_{b,c,x,y} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} + \beta_c \tag{3.7}$$

where ϵ is added for numerical stability.

3.4 Training a neural network

While classical methods need to repeat their parameter optimisation for every problem instance, a deep-learning model provides a generic representation of the problem, such that it can be applied to any problem instance without having to change its parameters. This internal representation is parameterised by the weights and biases of the hidden layers, which need to be optimised based on a large dataset of training samples. It is then assumed that this learned representation can be generalised to other unseen samples.

3.4.1 Loss function

A loss function quantifies how well the network performs in its task on a certain dataset. In case of supervised training, the input data and its corresponding ground-truth labels are available during the training process and can be used to define a loss function.

The loss function depends on the type of problem. For binary classification problems, for example, which have categorical outputs, a popular loss-functions is the cross-entropy loss. For regression problems with a continuous output values, the L1 or L2 norm can be calculated between the ground-truth values and their network predictions. Comparing multi-dimensional data, like images, can be done through the normalised cross-correlation loss, which measures the correlation between images X and Y as follows:

$$\mathrm{NCC}(X, Y) = \frac{1}{N-1} \sum_{i=1}^{N} \frac{(X_i - \mu_X)(Y_i - \mu_Y)}{\sigma_X \sigma_Y}. \tag{3.8}$$

with N the number of samples. Instead of evaluating the mean μ and standard deviation σ over the whole image domain, it can be calculated in a local neighborhood of 9-by-9 pixels around each point, yielding the local normalised cross-correlation loss **?**.

3.4.2 Back-propagation

During the training of the network, the loss function $\mathcal{L}$ is being optimised by the optimiser by tuning the network weights $\{W_i\}$ and biases $\{b_i\}$ proportional to how much they contribute to the loss function. Therefore it is important to know how each network weight depends on a change in the loss function, which mathematically translates into the derivative of the loss function with respect to the weights. Those derivatives are derived by the back-propagation algorithm **?**.

Assume, for the sake of simplicity, a two-layer network, where each layer takes the weighted sum of its input (eq.(3.1)) and applies an activation function f or g. A forward pass through the network can schematically be represented as follows:

$$X \to Z_1 = f(b_1 + W_1 X) \to \hat{Y} = g(b_2 + W_2 Z_1) \to \mathcal{L}(\hat{Y}, Y) \tag{3.9}$$

where X, $\hat{Y}$ and Y represent the network input, the network output and the expected output, respectively. The gradient of the energy function with respect to the weights of the second layer is given by:

$$\frac{\partial \mathcal{L}}{\partial W_2} = \frac{\partial \mathcal{L}}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial W_2} \tag{3.10}$$

$$= \frac{\partial \mathcal{L}}{\partial \hat{Y}} \frac{\partial g}{\partial (b_2 + W_2 Z_1)} Z_1 \tag{3.11}$$

Note that similar derivation can be done with respect to the bias parameter. The gradient with respect to the weights of the first layer can be calculated through the chain rule, as the first layer response is nested within the second layers response:

$$\frac{\partial \mathcal{L}}{\partial W_1} = \frac{\partial \mathcal{L}}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial Z_1} \frac{\partial Z_1}{\partial W_1} \tag{3.12}$$

$$= \frac{\partial \mathcal{L}}{\partial \hat{Y}} \frac{\partial g}{\partial (b_2 + W_2 Z_1)} W_2 \frac{\partial f}{\partial (b_1 + W_1 X)} X \tag{3.13}$$

Note that the two first factors from eq.(3.11) are repeated. In general, any additional layer will add an additional factor to the gradient. By saving the gradients of the last layers, the gradients of the first layers can efficiently be calculated. It is said that the loss-function is *back-propagated* through the network.

3.4.3 Optimiser

The goal of the optimiser is to minimise the training loss-function by changing the network weights. It therefore relies on the gradient descent principle, as introduced in eq.(1.7) for a classical optimiser. It estimates the gradient of the loss function for the current model state with respect to the network parameters $\boldsymbol{\theta}$ (i.e. weights and biases) and updates the parameters accordingly:

$$\begin{aligned} \boldsymbol{\theta}_t &= \boldsymbol{\theta}_{t-1} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{t-1}) \\ &= \boldsymbol{\theta}_{t-1} - \alpha \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_{t-1}), \end{aligned} \tag{3.14}$$

where the gradients are computed by the back-propagation of section 3.4.2.

One limitation of eq.(3.14) is the fixed learning rate or step size α. It controls how much the weights are changed for a certain change in loss function. If the learning rate is too high, the optimiser will not converge and keep overshooting. If the learning rate is too low, training might take very long and might end up in a local minimum. However, one would expect to need a large learning rate in the beginning of the optimisation and a smaller one while you get closer to the minimum. Adagrad, for example, divides the learning rate by the square-root of the cumulative sum of all the preceding gradients squared. This results in a monotonously decreasing learning rate. RMSProp on the other hand can have a decreasing or increasing learning rate depending on if the gradients remain consistent or if they change direction.

A second concern about eq.(3.14) is that updates are only based on the current model state and not on the previous states, which can result in high fluctuating updates. This can be solved by adding a momentum term which accumulates the preceding gradients to calculate a parameter update. If more updates are taken into one direction the momentum will increase while updates in dimensions which change direction will be suppressed. As a result the convergence is accelerated in the relevant direction and oscillations are reduced.

The Adaptive Moment Estimation (Adam) optimiser combines both, the adaptive learning rate and the momentum acceleration, by estimating the first and second moments of the gradient by a moving average:

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{t-1}) \tag{3.15}$$

$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)(\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{t-1}))^2 \tag{3.16}$$

with β_1 and β_2 being hyper-parameters of the optimiser which control the exponential decay rate of the average. The initialisation of $\boldsymbol{m}_t$ and $\boldsymbol{v}_t$ introduces a bias to the averaging, and can be removed by multiplying with an additional factor as proposed in the original paper **?**. Ignoring this factor, the parameter update according to the Adam-optimiser is given by:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \frac{\alpha}{\sqrt{\boldsymbol{v}_t} + \epsilon}\boldsymbol{m}_t, \tag{3.17}$$

with α the stepsize.

3.4.4 Data batches

As networks are typically trained on large datasets, it becomes unpractical or even impossible to load all the training data in memory at the same time to optimise the network parameters. Therefore the dataset is typically split up into several batches of data which are loaded in memory one by one during the training process.

The back-propagation, gradient descent and parameter update are applied to each individual batch separately. Also the batch normalisation of section 3.3.3 is performed on each batch instead of on the entire database. When all batches of the training dataset have been processed, we say that one epoch has been completed. The training is repeated over several epochs, similar to the different iterations in classical optimisation methods.

The split of the training data into batches implies that the gradient of eq.(3.14) is only calculated from a random subset of training data, which is an approximation to the

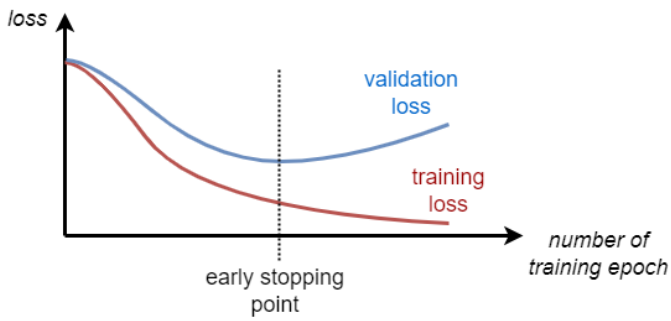


Figure 3.4: Typical evolution of the training and validation loss as function of the number of epochs.

real gradient that one would find for the entire database. Therefore, the optimisation scheme, as outlined in section 3.4.3, is also referred to as a stochastic gradient descent method.

3.4.5 Underfitting versus overfitting

The purpose of the network training is to find a model that minimises the training loss, but also generalises well to unseen input data. In order to evaluate the generalisability of the model during and after training, we typically rely on an additional validation and test dataset, respectively. In total, we have three datasets with a ratio of 80%-10%-10% in number of samples:

- The training dataset is used by the optimiser to optimise the network weights.
- The validation dataset provides an unbiased evaluation of the network after each epoch. This data is seen during training, but not used to learn from. The hyper-parameters of the model can be tuned based on the energy loss computed on this dataset.
- The test dataset provides an unbiased evaluation of the final model.

The final model should perform well on all three datasets. In the beginning of the training, the model under-fits all the datasets, and has a weak generalisability. As the training continues, both, training and validation loss will decrease as illustrated in Figure 3.4. The validation loss will eventually stagnate or even increase, while the training loss can further be decreased. At that point, the model starts over-fitting the training data. Instead of learning a generalisable representation of the data, it will exactly reproduce the training data after a while, including the noise. One way to prevent overfitting and to have the best generalisability, is by early stopping the training based on the validation loss **?**.

The overfitting/underfitting-problem is related to the fact that the number of parameters in a neural network needs to be in balance with the amount of training data. If the number of parameters is low compared to the number of training data, the model will underfit the data, meaning that it can not capture the underlying structure of the data well. This can easily be solved by increasing the model capacity by including more layers to the network. On the other hand, if the number of parameters is higher than the number of training data, it can overfit the data.

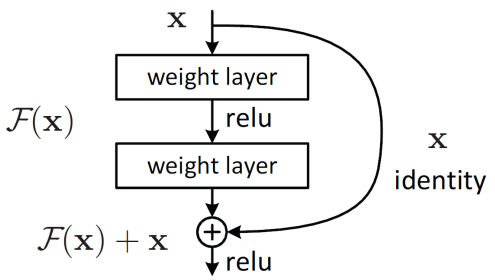


Figure 3.5: A residual block uses a shortcut connection to skip multiple layers in order to solve the vanishing gradient problem **?**.

Overfitting of the model to the training data can be prevented by so-called regularisation, which improves the generalisability of the network. The most common regularization approach is to add a penalty term to the loss function that prevents too large network weights and thus reduces the complexity of the decision boundary. The higher the regularization strength, the more overfitting gets reduced.

Another regularisation technique is to include dropout layers in the network, which turn on and off certain neurons in the network with a certain probability during training. This forces the network to learn features through different paths, resulting in weights to be better distributed over the network.

Finally, batch normalisation, discussed in section 3.3.3, also has a sort of regularization effect on the training. It normalises a layers output by subtracting the batch mean and dividing it by the batch standard deviation. The layer has also two additional learnable parameters that can shift and scale the data.

3.5 Examples

3.5.1 ResNet

After the general architecture of CNN being introduced, it is tempting to believe that deeper networks, with multiple layers, will be more successful, as they have more weights. However, adding too many layers leads to saturation of the training loss and eventually to an increase of the training error, a problem known as the vanishing gradient problem. Network weights are updated by back-propagating the energy function throughout the network, by using the chain rule as discussed in section 3.4.2. If succeeding gradients are small, the early layers in the network might not receive valuable updates, despite their importance at the beginning of the network.

To solve the vanishing gradient problem, deep-residual network (ResNet) was proposed, originally for image classification **?**. Instead of learning a non-linear mapping function $\mathcal{H}(x)$, it learns a residual mapping $\mathcal{F}(x) = \mathcal{H}(x) - x$, by introducing a skip connection which bypasses one or several layers. The residual function and the layers input are added at the end of the module, as shown in Figure 3.5. The skip connection makes it easier to learn an identity mapping between the input and output, which is a mechanism for the network to turn off irrelevant layers. The back-propagation can now also skip the intermediate layers thanks to the skip-connection, hence larger

gradients can reach the initial layers and all layers of a deep network can achieve a similar learning rate.

3.5.2 Encoder-decoder networks

Encoder-decoder architectures often arise in many deep-learning models and consist of two network parts. The first part, the encoder, compresses the input data into a "latent"-space variable with smaller dimensions. Its convolutional and max-pool layers reduce the spatial dimensions while increasing the number of features. The decoder part, on the other hand, decompresses the latent variable by upsampling or deconvolutions.

One example of an encoder-decoder network is the autoencoder, which is designed to reproduce its own input. The latent variable acts as a bottleneck between the encoder and decoder which prevents all features to be passed through. As a result the network can only output an approximation of its input image. This property makes such type of network for example interesting for denoising of medical images **?**.

The spatial downsampling in the encoder prevents the decoder of recovering the full spatial resolution. U-nets are designed to solve this issue by including skip connections between the encoder and decoder layers, as can be seen from Figure 3.6. These skip connections allow spatial information to flow from the encoder to the decoder, and can be concatenated with the feature information from the upsampling path of the decoder.

While U-nets were originally proposed for biomedical image segmentation **?**, their architecture has successfully been adopted in many other image processing applications. Registration networks, for example, deploy this U-shaped architecture with skip-connections to learn a deformation field from two input images **?**. Even a series of U-networks with feature aggregation between the different stages has been adopted for the sake of landmark detection **?**.

3.5.3 Style transfer

Neural networks are often trained on simulation data, such as DRRs for example (section 2.3). This simulation data might however still look very different from real data on which we want to apply the network. Augmentation of the training data, like varying the brightness, noise, etc are techniques to improve the generalisability of the network. However, it remains challenging to mimic the exact appearance of real data. Alternatively, the style difference between training data and real data can be learned by a so-called Generative Adversarial Network (GAN) **?**, by decoupling the content and style from the images. Given a real image, its style can be converted to the style of a simulation image, for which the network has been trained.

A GAN network is composed of a generator and discriminator which compete against each other. The generator is trained to create realistic-looking fake images based on a 1D latent space variable. The discriminator, on the other hand, is trained to distinguish between real and fake images. By training both at the same time, the generator is forced to create images as close as possible to the original ones, in order to fool the discriminator.

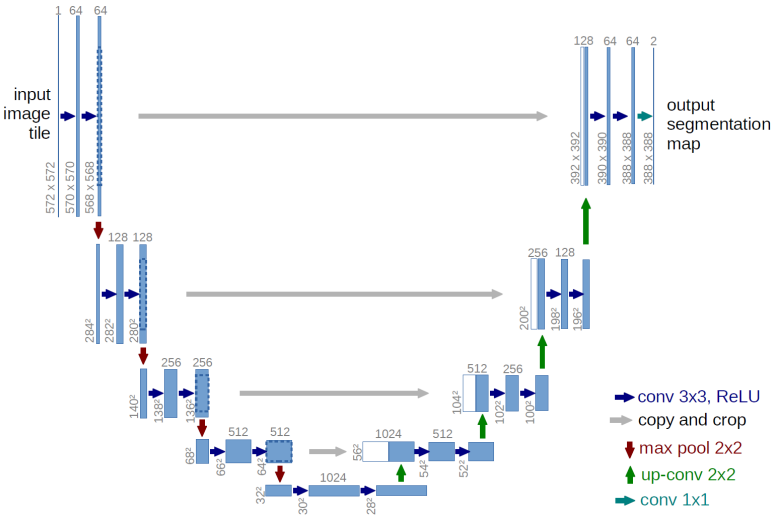


Figure 3.6: U-net architecture, consisting of an encoder and decoder with skip-connections in between **?**.

Part II

Contributions to surface registration of articulating bodies

4

Graphical User Interface for Joint Space Width Assessment by Optical Marker Tracking

Contents

Abstract

Optical position tracking is an essential tool in computer-assisted interventions for intra-operative guidance. It allows to register a pre-operative model or surgery plan to the patient, providing additional support to the surgeon. In this paper, we propose a two-step procedure to register pre-operative digital surface models to the surgical scene based on optical marker data. First a paired-point matching is applied, followed by an iterative closest point registration step. Mapping the surface model to the camera system allows to compute properties like the joint space width and motion asymmetry. Our method can be generalised to any joint and has been made available through an open-source graphical user interface, enabling future research on surgical navigation systems.

The work in this chapter has been published as:

J. Van Houtte, Sijbers, J., and Zheng, G., "Graphical User Interface for Joint Space Width Assessment by Optical Marker Tracking", in *4th International Conference on Bio-engineering for Smart Technologies*, 2021.

4.1 Introduction

Recently, many orthopaedic interventions have become computer-assisted in order to improve their clinical outcome and consistency **?**. Position tracking systems are playing a central role in such workflows as they allow to navigate surgical instruments relative to the patient. They allow to map a pre-operative plan to the patient intra-operatively. For anterior cruciate ligament reconstruction (ACL), for example, the tunnel position and direction has been optimised pre-operatively based on computed-tomography (CT) images **?**. For total knee arthroplasty (TKA) such pre-operative CT is used to select the right implant and to plan the best resections **?**. During surgery, the surgeon must be able to follow these surgical plans as close as possible.

Besides surgical navigation, position tracking is also relevant intra-operatively for evaluating kinematics during surgery. It has been indicated in the literature that restoration of the native knee motion in TKA, for example, leads to better clinical outcomes **?**.

An optical marker position tracking system is designed to measure the three-dimensional (3D) position of markers attached to a patient or to a surgical instrument. The stereoscopic system consists of two infrared (IR) illuminators and two camera sensors. The illuminator emits infrared light which gets reflected from the retroreflective coating of passive spherical markers, back to the camera sensor. Active markers on the other hand get activated by a trigger pulse emitted by the illuminator and emit IR light themselves towards the camera.

By arranging a set of several markers into a specific configuration on a surgical instrument, the position sensor is able to calculate the position and orientation of that instrument, in the form of a rigid transformation. This configuration of markers is referred to as the digital reference frame (DRF) and its design has been studied in the literature to maximise the tracking accuracy **?**. A single-face DRF has often a co-planar arrangement of four markers, with a large distance between them to avoid occlusions.

The tracking system computes the rigid transformation between the camera coordinate system and each DRF. It remains a question for the user how the bone is positioned with respect to this DRF. Different procedures exist for registering a pre-operative digital bone model to the DRF. Either a pre-operative CT is acquired with the markers already attached, such that the markers can directly be registered to each other without further refinement **?**. Such CT-acquisition, however, complicates the surgical workflow. An alternative approach utilizes a digital probe to annotate physical landmarks on the bone, which can be brought into correspondence with landmarks on the digital surface models. This is however sensitive to the subjective placement of the landmarks.

In this paper, we outline a two-step procedure to register a pre-operative digital surface model to optical marker data, consisting of a landmark registration and an iterative closest point registration. We demonstrate how the registered surfaces can be used to efficiently evaluate motion asymmetry in the knee joint. Our procedure has been made available through a graphical user interface (GUI), and is open-source available[1].

[1] https://github.com/jvhoutte/MarkerTracking

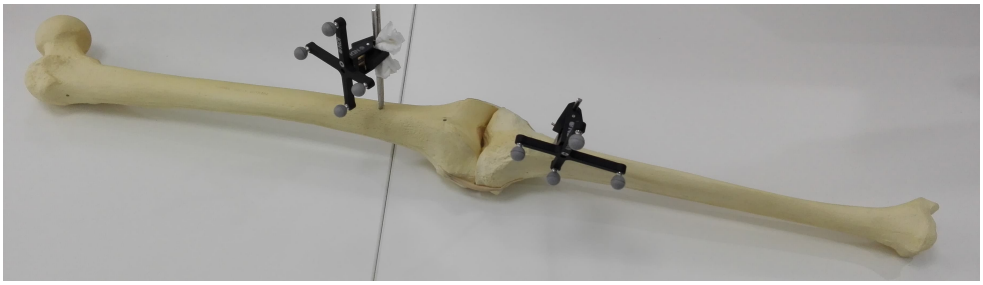


Figure 4.1: Plastic model of femur and tibia with optical markers attached.

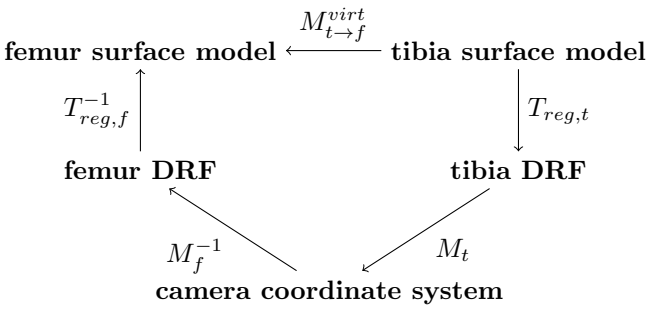


Figure 4.2: Schematic overview of the transformations between different coordinate systems. The position tracking system measures the transformations M_b between the camera coordinate system and the digital reference frame (DRF) of bone b. The calibration solves for the registration transformation $T_{reg,b}$ between the bone's surface model and the DRF.

4.2 Methodology

4.2.1 Data acquisition

In our experiment, we used a Polaris Vega optical tracking system from Northern Digital Inc. (NDI) to track the articulating motion of a plastic knee joint model **?**. This tracking system has a 3D positioning accuracy of 0.045 mm **?**. Digital reference frames with four reflective markers each were screwed into the femur and tibia bone as shown in Figure 4.1. The DRFs make it possible for the tracking system to indirectly track the bones. The plastic bones were manually articulated during the marker tracking. A CT scan was acquired from the plastic bone models in order to obtain a digital surface model after segmentation.

4.2.2 Calibration

The goal of the calibration step is to find the transformation $T_{reg,b}$ that registers the digital surface model of bone b to the corresponding digital reference frame defined by the reflective markers. As the position of the real bones with respect to the DRF is unknown, we sample the real bone surfaces by a digital annotator equipped with another digital reference frame. The optical tracking system simultaneously tracks the bone DRFs and the annotator's DRF. We denote the tip of the digital annotator

Figure 4.3: Registration of the tibia (top) and femur (bottom) digital surface model to their corresponding optical surface samples (red dots), after motion-correction. The white surface is the initialisation result after paired point registration. The blue surface is the result after ICP-registration.

by $L^{real}(t)$ and the tracked transformation of the bone's DRF by $M_b(t)$. Figure 4.2 summarizes the relationship between the different transformations.

In a first step the annotator is used to annotate a small number of physical landmarks on each bone b. Those landmark positions $L_b^{real}(t)$ are measured by the optical tracking system with respect to the camera coordinate system. To correct for any possible movement of the bones during the annotations, we compute the landmark positions with respect to the bone's DRF, which is given by: $M_b^{-1}(t)L_b^{real}(t)$, with $M_b(t)$ the bone's DRF transformation.

The same set of landmarks is then annotated on the digital surface model of the bone and a paired point matching algorithm between both sets of landmarks is performed. The resulting transform T_{init} aligns the landmark positions on the surface model with the real landmark positions in the bone's DRF.

In a second step, the digital annotator is moved along the bone's surface in order to acquire a continuous set of digital points S. Next, we apply a dense surface matching by iterative closest points (ICP) between the digital surface model and the sampled points on the real surface $M_b^{-1}(t)S_b(t)$. The resulting transform T_{icp} brings the digital surface model into closer correspondence with the real model after the initialisation. The final registration transform for each bone is given by:

$$T_{reg} = T_{icp}T_{init} \tag{4.1}$$

An example of both registration steps is shown in Figure 4.3. It qualitatively shows the necessity of the last dense registration step, to obtain close registration with the continuous set of motion-corrected points (red dots in the figure). The initial transformation, resulting in the white surface in the figure, is sensitive to subjective landmark placement and only provides a rough alignment. Note that the final

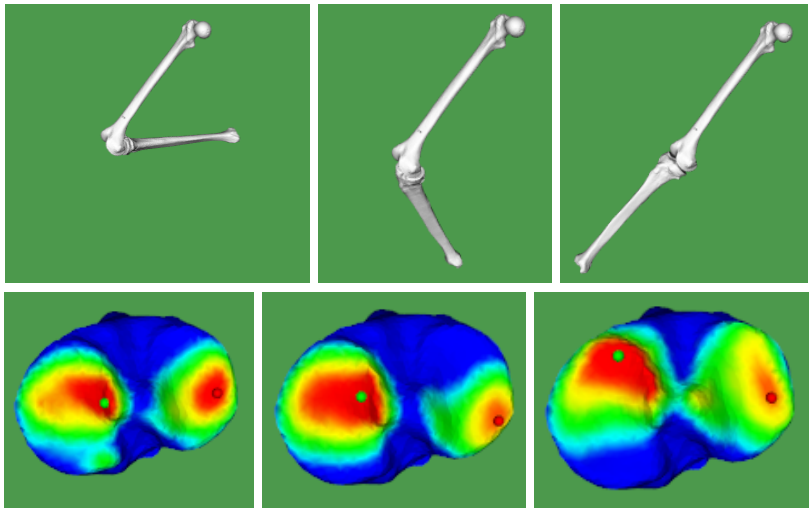


Figure 4.4: Example of reconstructed poses (top row) with the corresponding joint space width shown on the tibia joint surface by the color scale (bottom row). The green and red dot indicate the location of the minimal joint space width for the medial and lateral side, respectively.

transformation T_{reg} is time independent. It is used to position the digital surface model correctly with respect to the bone's digital reference frame.

4.2.3 Articulated transformations

Combining the calibration transformations T_{reg} and the time-dependent motion transformations $M(t)$, computed by the optical tracking system, the digital surface model can be transformed to the camera coordinate system by:

$$M(t)T_{reg} \tag{4.2}$$

As the joint space width is invariant to global transformations, we choose, without loss of generality, to keep the femur model fixed and transform the tibia relative to the femur by:

$$M^{virt}_{t\to f}(t) = T^{-1}_{reg,f} M_f^{-1}(t) M_t(t) T_{reg,t} \tag{4.3}$$

4.2.4 Asymmetry of joint space width

The previous steps allow to efficiently calculate the joint space width during articulation. First, an implicit surface distance function $D(.)$ is calculated for the femur surface model, as illustrated in Figure 4.5. This function is defined within a region of interest $\Omega \subset \mathbb{R}^3$, centered around the femur condyles. It computes the distance from each point $\boldsymbol{x} \in \Omega$ to the closest point on the femur surface model. The calculation of the distance function $D(.)$ is the time-consuming step but only needs to be performed once as the femur stays fixed in space at all times. Given a point $\boldsymbol{p}$ on the tibia surface model, the closest distance d to the femur at a particular time t can be found by the fast evaluation of $D(.)$:

$$d = D(M^{virt}_{t\to f}(t)\boldsymbol{p}), \tag{4.4}$$

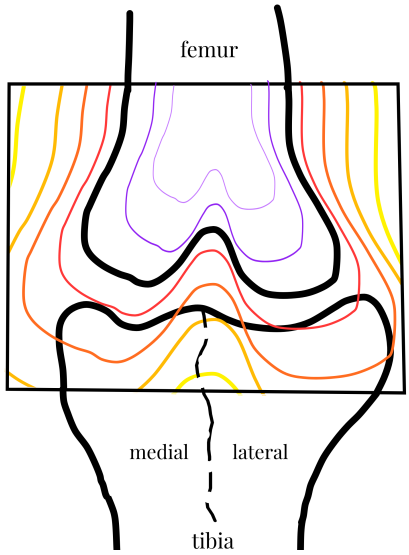


Figure 4.5: Illustration of the implicit surface distance function of the femur, which calculates the distance of every point inside the region of interest to the closest point on the femur surface. The function is evaluated for the points on the tibia model, discriminating between the medial and lateral side of the tibia.

with $M_{t\to f}^{virt}(t)$ given by eq.(4.3). Figure 4.4 shows the joint space width on the tibia joint for different poses.

In order to study the asymmetry of the joint movement, we make a distinction between the joint space width at the medial and lateral side of the tibia. Both sides are automatically identified by calculating the symmetry plane of the tibia as follows. First the elongation axis of the bone is calculated by applying PCA on the surface model and extracting the eigenvector with the largest eigenvalue. This eigenvector corresponds to the elongation axis and can be aligned with the y-axis, without loss of generality. Next, the model is mirrored with respect to the xy-plane. The original and mirrored models are registered to each other by ICP. The previous steps are repeated for different roll rotation angles around the y-axis, because ICP is vulnerable to local minima. We select the registration result with the smallest geometric error. The centers of the lines connecting the model points with their mirrored counterparts lie approximately on one plane. We apply a plane fit on this set of points to obtain the symmetry plane.

4.3 Discussion

This paper described a two-step calibration procedure in which a pre-operative surface model can accurately be registered to markers detected by an optical position tracking system. We illustrated how, after calibration, the mapped surfaces can be used to efficiently calculate the joint space width during articulation.

Our method has been made publicly available through a graphical user interface, shown in Figure 4.6. It allows to load any pair of bones. After loading the surface

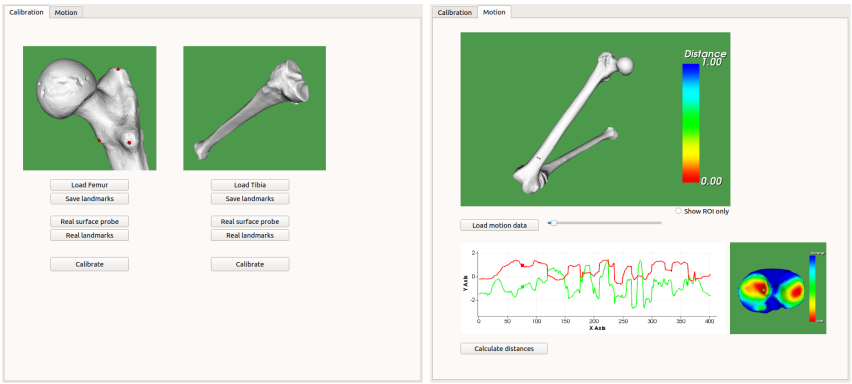


Figure 4.6: Screenshot of the graphical user interface. The first window lets the user load the data and annotate the landmarks on the digital surface models. The second window displays the articulating surface models and the minimal joint space width in function of the time.

models and the position tracking data, the user is asked to annotate the landmarks on the digital surface models. In the second window the transformed surface models are visualised at each time frame. A graph shows the minimal joint space width for the medial and lateral sides as function of the time.

While the proposed GUI is valuable for accurate point cloud registration and motion asymmetry studies, there is still room for improvement in terms of marker usage in the acquisition protocol. Marker-less navigation systems have been proposed to avoid invasive marker placement. Such systems simultaneously use RGB and depth cameras **?**. Based on the RGB images, a (deep-learning) segmentation method can extract a segmentation mask of the knee joint. This mask is subsequently used to extract the region of interest from the point cloud acquired by the depth camera. A pre-operative surface model can be registered to this partial point cloud by means of ICP in order to establish the relative pose. This workflow omits the use of markers and sampling of the bone by a digital probe, but requires the knee joint always to be completely visible. The occurrence of occlusions during full articulation of the knee makes this method less suitable, favoring marker-based tracking methods.

Both, marker-based and marker-less tracking systems, require the registration of a surface model to a point cloud. Only the acquisition method to obtain this point cloud might differ. It remains open for further research how both can be combined. Marker-less systems might, for example, benefit from an initialisation provided by a digital annotator.

4.4 Conclusion

This paper proposed a two-step procedure to register surface models to optical marker tracking data, consisting of a landmark-based and dense registration. The accurately mapped surface models allow to study the asymmetry of the joint space width in a computationally efficient manner. The method has been made available through an open-source graphical user interface in order to support future research.

5

An Articulating Statistical Shape Model of the Human Hand

Contents

Abstract

This paper presents a registration framework for the construction of a statistical shape model of the human hand in a standard pose. It brings a skeletonized reference model of an individual human hand into correspondence with optical 3D surface scans of hands by sequentially applying articulation-based registration and elastic surface registration. Registered surfaces are then fed into a statistical shape modelling algorithm based on principal component analysis. The model-building technique has been evaluated on a dataset of optical scans from 100 healthy individuals, acquired with a 3dMD scanning system. It is shown that our registration framework provides accurate geometric and anatomical alignment, and that the shape basis of the resulting statistical model provides a compact representation of the specific population. The model also provides insight into the anatomical variation of the lower arm and hand, which is useful information for the design of well-fitting products.

The work in this chapter has been published as:

J. Van Houtte, Stanković, K., Booth, B. G., Danckaers, F., Bertrand, V., Verstreken, F., Sijbers, J., and Huysmans, T., "An Articulating Statistical Shape Model of the Human Hand", in *Advances in Human Factors in Simulation and Modeling (AHFE 2018)*, Cham, 2019, vol. 780, pp. 433–445.

5.1 Introduction

Shape models of faces and full-bodies have become valuable for many commercial applications of computer vision and graphics, ranging from customized design to motion tracking **??**. Their potential for noise and artifact reduction, hole filling, and resolution improvement, have aided to employ low-budget scanners with low mesh quality **??**. Recently, the popularity of these techniques has led to their consideration for modelling the human hand, most often for the task of hand tracking **??**.

In the context of hand tracking, shape models have been used with the primary goal of improving pose estimation **?????**. In general, these techniques consist of a fixed prior rigged template model which can be aligned to person-specific depth images or 3D meshes. The alignment is often achieved by solving for the articulation and anthropometric parameters of the template that optimally match the subject's depth image or 3D mesh. The registration is regularized by principal component analysis (PCA) **?** or by an "as rigid as possible" (ARAP)-regularization **?**.

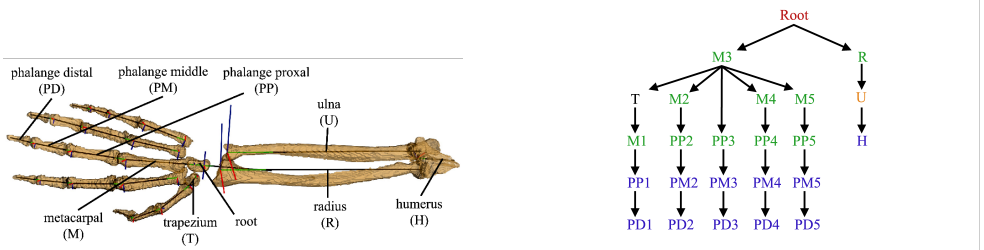
As the focus of these techniques has been on obtaining accurate pose information, the level of geometric detail can vary significantly between models. Many models are composed of primitives like spheres and cylinders of fixed size, with the registration step simply articulating these primitives **??**. Others use a more realistic skin geometry, but only allow the model to articulate **?**. Accommodating variations in hand shape and size has only recently been explored **??**, and those variations have not been restricted to a range of "natural" hand shapes and sizes.

It has been argued that detailed personalized hand models improve the accuracy of both model registration and pose estimation **?**. This argument was furthered by Khamis et al. who regularized possible hand shapes with a low-dimensional parametric shape model that included statistical shape variations of a population **?**. Ideally, this shape model would be based on a dataset of high-quality surface scans in the same pose, but Khamis et al. constructed their shape basis on low-quality depth scans which contained self-occlusions. To address the low quality, their statistical shape model was estimated simultaneously with each individual's hand shape and pose parameters.

Meanwhile, recent advances in optical scanning technology, such as the 3dMD-system **?**, have enabled the acquisition of high-quality ($< 0.5\,\text{mm}$ error) 3D surface scans, even from highly articulating objects like hands. It is expected that a statistical shape model based on these high-quality scans would reveal more geometric details and it is therefore the interest of this paper to build a high-resolution geometric shape basis that, to the best of our knowledge, has not been seen in the literature.

Building such as statistical model requires bringing the 3D scans of different subjects' hands into anatomical correspondence. Having an anatomical correspondence for all points of all meshes is critical to build an accurate and interpretable model. However, this is an especially difficult task for a complex articulating shape like the hand.

The aim of this study is to obtain reliable anatomical correspondence for building a statistical human hand model. We propose a registration algorithm, similar to the technique in **?**, that aligns a template articulation model to a database containing 100 high-quality 3D scans of human hands. We hypothesize that the addition of the articulation model, and its corresponding registration algorithm, will allow us to



(a) Arm skeleton, with labelled bones

(b) Bone hierarchy

Figure 5.1: Our reference articulating hand model is defined by the skeleton in (a). The bones in this skeleton are ordered in the hierarchical tree structure in (b) with an artificial root bone at the wrist. Arrows indicate the parent-child relationship. Colors indicate the corresponding articulation parameters: α(blue), α and β(green), γ(red), δ(orange). See text for further details.

more accurately obtain shape correspondences, and normalize for pose, in 3D scans of human hands. We further hypothesize that these advances in shape correspondences and pose normalization will facilitate the use of standard statistical shape modelling algorithms, like PCA, on 3D scans of human hands.

5.2 Methods

At a high level, our proposed shape modelling technique works as follows. An articulating reference of the human hand, with anatomically correct rotation axes, angles, and constraints, is constructed to act as prior in an articulation-based registration method. This reference hand is then registered to each optical surface scan of a database in order to make anatomically correct correspondences between them. Person-specific deviations that cannot be captured by the reference are accommodated through a subsequent elastic surface registration step. Finally, the registered surfaces are articulated to the same pose and PCA is used to derive the statistical shape model of the human hand. The following subsections discuss these steps in further detail.

5.2.1 Reference Articulating Hand Model

5.2.1.1 Reference surface geometry and skeleton

Our reference hand is based on a single Magnetic Resonance Image (MRI) scan of the first author's right hand (repetition time [TR]: 4220 ms; echo time [TE]: 1560 ms; field of view [FOV]: 192 mm $\times$ 520 mm; resolution: 1 mm^3; no gap). The outer skin surface and all relevant bones were manually segmented from the MR image.

To construct the surface mesh of the reference hand, the binary label field of each body part, obtained from the MRI scan, was then converted to a triangulated surface mesh using a discrete marching cube algorithm **?**. The extracted skin surface mesh was then removed of noisy outliers, smoothed in volume-preserving way **?** and remeshed uniformly **?**. The reference skin mesh is denoted by $\hat{M}_M = (\hat{V}_M, \epsilon_M)$, with $\hat{V}_M \in R^{3\times N_M}$ a matrix containing the coordinates of the N_M vertices in a rest pose (the rest pose being denoted by the hat), and $\epsilon_M \in R^{N_M \times N_M}$ representing the connectivity, which remains constant at all times.

An abstract line-skeleton $\hat{S}$, defined using the set of segmented bones, is shown in Figure 5.1a. Each segmented bone b is represented by a local coordinate frame in the skeleton (i.e. an origin and orientation). The origin of the bone is located at its center-of-rotation $\boldsymbol{h}_b$ and the orientation of its coordinate frame is as described by the International Society of Biomechanics (ISB) **?**. The orientation of each bone with respect to the world reference frame is described by the bone-to-world rotation matrix $C_b \in SO(3)$.

5.2.1.2 Bone Hierarchy

The set of bones are ordered in the hierarchical tree structure shown in Figure 5.1b. This hierarchy represents the parent-child relationships between the coordinate frames of each bone in the skeleton. The root of the hierarchy is an artificial bone located at the wrist with the same orientation as the third metacarpal bone. A wrist-rooted armature allows us to describe arm and hand motion independently from each other, but with respect to a common root coordinate system at the wrist (this decoupling will be a benefit in our registration tasks). Global hand motion is described by the third metacarpal bone, which the ISB standard defines as the parent of all other carpal bones **?**.

5.2.1.3 Articulation

The articulation of the hand is defined by the state of its joints. Using our skeleton, the state of each joint can be described as a rotation between the local coordinate frames of adjacent bones:

$$R_b = C_{p(b)} C_b^{-1} \tag{5.1}$$

where $p(b)$ is the parent of bone b as defined by the tree structure in Figure 5.1b. The rotation matrix R_b captures how bone b is articulated with respect to its parent. This matrix can be decomposed into three rotation angles, $\alpha_b, \beta_b, \gamma_b$, which match the ISB's joint angle descriptions **?**. The angle α_b is the primary angle of articulation and describes the bending of the fingers and flexion/extension of the wrist. The angle β is the secondary angle of articulation and describes ulnar/radial deviations of the wrist and the separation between the fingers. The angle γ is a roll angle around the bone's longitudinal axis. An additional angle, δ, is used to define pronation-supination of the arm. This motion is modelled as a rotation around an axis connecting the ulna at the wrist to the radius at the elbow. In the wrist-centered armature the ulna rotates around this axis over the radius. The degrees of freedom for each bone have been indicated by the color in Figure 5.1b: α(blue), α and β(green), γ(red), δ(orange).

When articulating a bone with a new set of angles, we recalculate the parent to bone rotation matrix as a concatenation of these rotation angles. From eq.(5.1) it is possible to update the rotation matrix R_b since its parent maintained the same position in space. Relating the rotation matrix of the rest pose with this of the articulated pose, provides the rest-to-pose rotation matrix:

$$T_b = C_b \hat{C}_b^{-1} \tag{5.2}$$

from which we can update the head position of the bone and update the bones further down in the tree hierarchy.

Finally, we confine, by visual inspection, all joint articulation angles to remain within a natural range of motion. To accomplish this, we introduce a mapping from these constrained physical angles to "dummy" unconstrained variables as described in **?**. The benefit of the "dummy" unconstrained variable is that it can be optimized in the registration algorithm without any changes to the optimizer.

5.2.1.4 Anthropometric scaling

Besides the articulation of the skeleton, the reference model also accommodates the anthropometric variations related to bone length and body part thickness. The model therefore adopts an affine scaling of each bone defined by a longitudinal scaling factor $s^{\parallel}$ and a transversal scaling factor $s^{\perp}$. The scaling matrix in world coordinates can be written as follows:

$$S_b = C_b \, \mathrm{diag}(s_b^{\perp}, s_b^{\parallel}, s_b^{\perp}) \, C_b^{-1}. \tag{5.3}$$

The world to bone transformation including both articulation and scaling is therefore defined as:

$$F_b = S_b T_b. \tag{5.4}$$

In the reference hand, we apply longitudinal and transversal scaling on the lower arm, hand palm, and each finger separately. For the fingers, a single longitudinal scaling factor is used for all phalanges of the same digit; this is justified by the fact that the ratio of bone lengths between phalanges of a single digit obey closely the golden ratio rule **?**. Nevertheless, we allow the metacarpals to change in length independently from the phalanges in order to maintain flexibility of the reference during the registration task.

5.2.1.5 Skinning

To deform the reference skin mesh $\hat{M}_M$ to a new skin mesh $M_{M(\Phi)}$ in line with the articulation parameters Φ of the skeleton, we employ Linear Blend Skinning (LBS) **?**. LBS updates vertices based on the skeleton's articulation via:

$$\boldsymbol{v}_i = \sum_b w_{i,b} F_b \hat{\boldsymbol{v}}_i + \boldsymbol{t}_b, \tag{5.5}$$

with $\boldsymbol{t}_b = \boldsymbol{h}_b - F_b \hat{\boldsymbol{h}}_b$ being a translation vector. The skinning weights $w_{i,b}$ capture how much vertex $\boldsymbol{v}_i$ is influenced by articulating bone b. They are obtained by solving a heat equilibrium analogy as described in **?**.

During the pronation-supination movement of the lower arm, the amount of skin sliding gradually increases over the elongation axis of the arm. This twisting behavior cannot be explained with standard linear blend skinning since the expected skin deformation does not follow the transformation of its underlying bone. Instead, we model the skin deformation during pronation-supination by applying spherical linear interpolation (SLERP) **?** between C_{ulna} and C_{radius}, where the interpolation parameter t linearly increases from the ulna's head at the wrist to the radius' base at the elbow **?**. A vertex is then rotated with the interpolated rotation matrix depending on its location along the connection axis.

$$\boldsymbol{v}_i = [w_{i,ulna} t T_{ulna} + w_{i,radius}(1-t) T_{radius}] \hat{\boldsymbol{v}}_i. \tag{5.6}$$

Group Φ_j	Level $\{\phi_i\}$	Degrees of freedom	Relevant bones $B \subset S$
HAND	A	α_{M3}, β_{M3}	M2-5, PP2-5
	B	α_{PP2-5}, β_{PP2-5}	PP2-5
	C	γ_{root}, α_{M3}, β_{M3}	M2-5, PP2-5
ARM	A	α_R, β_R	U, R
	B	δ_U	H
	C	α_H	H
SCALING	A	$s^{\perp}_{U,R,H}$, $s^{\perp}_{M1-5,PP1-5,PM2-5,PD1-5}$	U, R, M2-5, PP2-5
RIGID	A	global translation and rotation	Root, M5
THUMB	A	α_{M1}, β_{M1}, α_{PP1}	PP1, PD1
	B	α_{PD1}	PP1, PD1
	C	α_{M1}, β_{M1}, α_{PP1}, α_{PD1}, $s^{\parallel}_{M1,PP1,PD1}$, $s^{\perp}_{PP1,PD1}$	PP1, PD1
FINGER *	A	α_{M*}, β_{M*}, α_{PP*}, β_{PP*}, α_{PM*}, α_{PD*}	PP*, PD*
	B	α_{PP*}, β_{PP*}, α_{PM*}, α_{PD*}, $s^{\parallel}_{M*}$, $s^{\parallel}_{PP*,PM*,PD*}$, $s^{\perp}_{PP*,PM*,PD*}$	PP*, PD*

Table 5.1: Parameter hierarchy. The parameter set is divided in independent parameter groups Φ_j. Parameters in each group are organised in different levels, where each level is optimised at a time. Optimisation is done iteratively between levels within each group.

5.2.2 Articulation-based Registration

5.2.2.1 Hierarchical optimization

The aim of this section is to fit the articulation model, described in the previous section, to a 3D surface scan, denoted by $M_T = (V_T, \epsilon_T)$. This registration is done by optimizing a set of model parameters Ω via a non-linear Levenberg-Marquardt (LM) optimization scheme. The parameters include the articulation parameters, anthropometric scaling parameters, and rigid transformation parameters, summarized in Table 5.1. To avoid the optimizer ending in local minima, we subdivide the set Ω in several smaller groups of parameters $\Omega = \{\Phi_j = \{\phi_i\}_j\}$ and order the parameters in each group Φ_j in a hierarchical structure which is optimized iteratively (e.g. A, A-B, A-B-C) by the LM optimizer. By using a wrist-centered armature, we can decouple the hand and arm related parameters and do their registration steps independently. The order in which we optimize the defined parameter groups are: “hand”, “arm”, “rigid”, “scaling”, “rigid”, “hand”, and “arm”. Furthermore, we optimize each finger independently.

5.2.2.2 Landmark-based initialization

Before starting the hierarchical iterative optimization protocol, we initialize the registration by globally scaling and aligning the reference hand based on three landmarks: two at opposite sides of the wrist and one at the middle fingertip. Additionally, the length of the arm is set based on the distance between landmarks at the wrist and an additional landmark at the elbow pit. This second step was performed due to missing elbow geometry in our scan dataset, and would not be required if the elbow is thoroughly scanned.

5.2.2.3 Energy function

At each hierarchy level, we apply a LM optimization to minimize:

$$\phi = \arg\min_{\phi} \left[\sum_{i=1}^{|V_M|} w_a(i, B) \left| \min_{j} \left(d(\boldsymbol{V}_{M(\phi)}(i), \boldsymbol{V}_T^{\parallel}(j)) \right) \right|^2 \right], \tag{5.7}$$

where w_a is a binary weight used to turn on and off the contribution of vertices, depending on whether its corresponding bone is in the set of bones B considered to be relevant for the optimisation (see Table 1). $V_T^{\parallel}$ is the subset of V_T consisting of vertices whose normals are within 72° from the normal at $\boldsymbol{V_M(i)}$ (a more strict threshold of 37° is used for the scaling and arm optimisation steps). Rather than excluding points based on their normals, we search for the closest point that meets this normal angle condition. By doing so, we ensure that all points on the mesh will have a corresponding point (as long as the mesh is not too sparse). Points for which a counterpart was not found are excluded from the energy function.

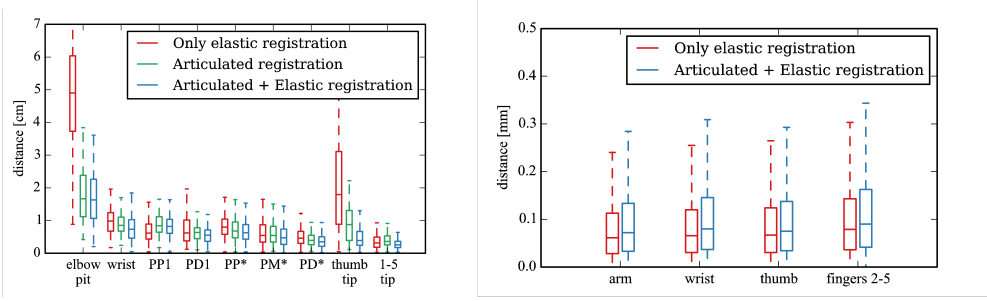
The distance measure $d(\boldsymbol{p}, \boldsymbol{q})$ used is the point-to-plane distance introduced by Park and Subbarao **?**. This is beneficial over point-to-point distance when using low resolution mesh, but cannot be used for optimizing arm supination since corresponding reference and target vertices lie in the same plane. In that situation, we replace the distance measure by its point-to-point variant.

5.2.3 Shape Correspondences

Initially, the vertices in our 3D meshes are randomly ordered, meaning that, say, vertex $\boldsymbol{v_i}$ in our reference mesh does not anatomically correspond to vertex $\boldsymbol{v_i}$ in another hand mesh. The number of vertices may also be different for every mesh. Before performing statistical analysis on these meshes, we must first establish an anatomical correspondence between them. This correspondence is achieved in two steps. First, the articulation-based registration, described above, is performed to align our reference hand to the target mesh. Second, an elastic registration algorithm is applied to provide a more precise anatomical correspondence between the reference mesh and the target mesh **?**. The final result is that the reference surface is deformed to have its shape as similar as possible to the shape of the target surface. At this point, the target mesh is replaced by the deformed reference, ensuring that each hand mesh has the same number of vertices ordered in the same fashion. This consistent vertex order ensures that every hand mesh has the same vertices in the same anatomical positions.

5.2.4 Pose normalization

In the statistical model, we are only interested in anthropometric variations and want to normalize as much as possible for any variation due to pose and articulation differences. Therefore, we apply a pose normalization on the elastically deformed mesh, using the skeleton estimated by articulation-based registration. Pose normalization can easily be achieved by interchanging the rest and pose articulations, i.e. inverting the rest to pose transformation matrix in eq.(5.2). Finally, all pose-normalized scans are centered around their center-of-gravity position.



(a) Landmark correspondence error

(b) Geometric error

Figure 5.2: The anatomical and geometrical correspondence results for our registration method and a purely elastic method. Anatomical correspondence error, using expert-denoted landmarks, is shown in (a) while geometric errors in the hand shapes are shown in (b).

5.2.5 Shape Modelling

To investigate the principal modes of variations present in the population, we apply a linear dimensionality reduction algorithm on the pose normalized registered scans. A popular choice for statistical shape modelling is a principal component analysis (PCA) **?**. In our context, PCA converts the vertex sets from all meshes into smaller sets of values through the definition of linearly uncorrelated variables called principal components. These principal components are defined by applying an orthogonal transformation on the original vertex coordinates. The position of vertex $\boldsymbol{v_i}$ in the statistical model is modelled as its average position $\boldsymbol{\mu_i}$ plus a linear combination of principal components $P_{i,j}$:

$$\boldsymbol{v_i} = \boldsymbol{\mu_i} + \sum_j \boldsymbol{w_j} P_{i,j} \tag{5.8}$$

The weights $\boldsymbol{w_j}$ give the contribution of each principal component (PC) to the model instance. The calculated PCs describe orthogonal directions of variance and they are ordered based on the fraction of variance found along the direction.

5.3 Results

In this section, we provide the results of the proposed registration and model-building techniques after testing them on a set of 100 static optical surface scans acquired with a 3dMD system. For comparison purposes, we also applied the elastic registration on the dataset as described in section 5.2.3 but without the articulation-based initialization of section 5.2.2.

5.3.1 Articulation-based Registration

5.3.1.1 Anatomical Correspondence

To quantify the anatomical accuracy of the registration method, we annotated 22 anatomical landmarks on the reference mesh and on each target scan. Landmarks were annotated at anatomical feature locations: at the elbow pit, at two opposite points around the wrist, at each fingertip and at all finger joints. We calculated the

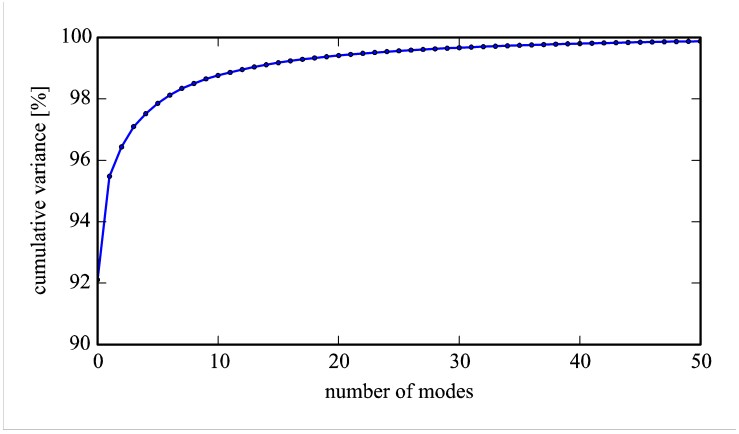


Figure 5.3: Normalized compactness graph of the statistical hand shape model.

distance between the landmark positions on the moving mesh and their ground-truth counterpart on the target mesh. These distances were computed after our articulation-based registration, after our elastic registration, and for the result of a purely elastic registration, without articulation-based initialization.

The landmark correspondence results are shown in Figure 5.2a. The average distance between joint landmarks after articulation and elastic registration was 5.7 mm, compared to 6.8 mm without the articulation based initialization step. Anatomical alignment is the best at the fingertips and distal joints because its estimation relies on clear geometric features. The accuracy on the elbow pit alignment is low due to missing data at the elbow and limited geometry information at the upper arm. Given the improved landmark correspondence of our algorithm, we can conclude that the articulation-based registration - as an initialization step - improves the anatomical correspondence of the elastic registration.

5.3.1.2 Geometric Correspondence

To create shape correspondence, we replace a target mesh by the registered result. This step may introduce geometric error where the surfaces do not match exactly. We quantify this geometric correspondence accuracy by calculating the average distance between the target and the elastically registered mesh, in the normal direction on the registered mesh. The results are shown in Figure 5.2b, with the average distance between surfaces grouped by anatomical region. The average geometric accuracy of our algorithm was 0.12 mm.

5.3.2 Statistical model

5.3.2.1 Model Performance/Compactness

The compactness of a statistical model is a widely used measure to quantify how efficiently the model describes the total variance in the population **??**. The compactness measure $C(m)$ is defined as the sum of the shape variance captured by the first m

principal components:

$$C(m) = \sum_{i=1}^{m} \lambda_i, \tag{5.9}$$

with λ_i the shape variance described by the i^{th} PC. Figure 5.3 shows the normalized compactness results of our statistical hand shape model. The first principal component explains over 90% of the total variability in the dataset, while the first four PC account for over 97%.

Our model's first four principal components are visualized in Figure 5.4. The average geometry is shown along with +/- three standard deviations for each principal component. The first PC describes global scaling. The second PC describes variations in the length-to-thickness ratio of the arm, hand and fingers. The third and fourth PC are related to varying length and width of the fingers relative to arm size, respectively.

5.4 Discussion

We have presented a two-step registration method for 3D meshes of human hands. First, we matched an articulating prior model to a target scan, then we applied an elastic registration to obtain more precise shape correspondence. We demonstrated our method on a dataset of 100 optical 3D surface scans. We showed that the anatomical accuracy improves by 17% by initializing the elastic registration with the articulation-based registration result, while the average geometric accuracy stays around 0.12 mm. We further fed the registered surfaces into a statistical shape modelling algorithm and showed that the resulting model provides a compact representation of the population's variation. Only four principal components are needed to describe 97% of the shape variability in the dataset. We believe that our model is suitable for applications like hole-filling and resolution improvement, where pose estimation is an inevitable task. Our shape model could also be useful as a prior in a surface registration algorithm.

Nevertheless, our results did highlight a few limitations. We observed low accuracy on the estimation of the elbow pit location mainly due to missing data around the elbow and limited geometry at the upper arm. We also noted that the registration outcome highly depends on its settings (e.g. the ranges of motion, order of parameter optimizations, vertex normal thresholds). Finally, it is likely that some articulation information did make it into the shape model as a result of errors in the articulation-based registration. The source of these errors include the optimization settings, but also the limited degrees of freedom in the reference hand (e.g. the use of the golden ratio to scale finger bones). Our future work will look at addressing these limitations as well as extending the technique to the 4D modelling of hand motion.

5.5 Conclusion

We presented herein a registration method for 3D meshes of human hands. It was based on the alignment of an articulating reference hand and elastic deformation. We demonstrated the registration's effectiveness by building a PCA shape model of the human hand. In the future, we will improve the anatomical accuracy of the methodology and to extend the method to model hand motion.

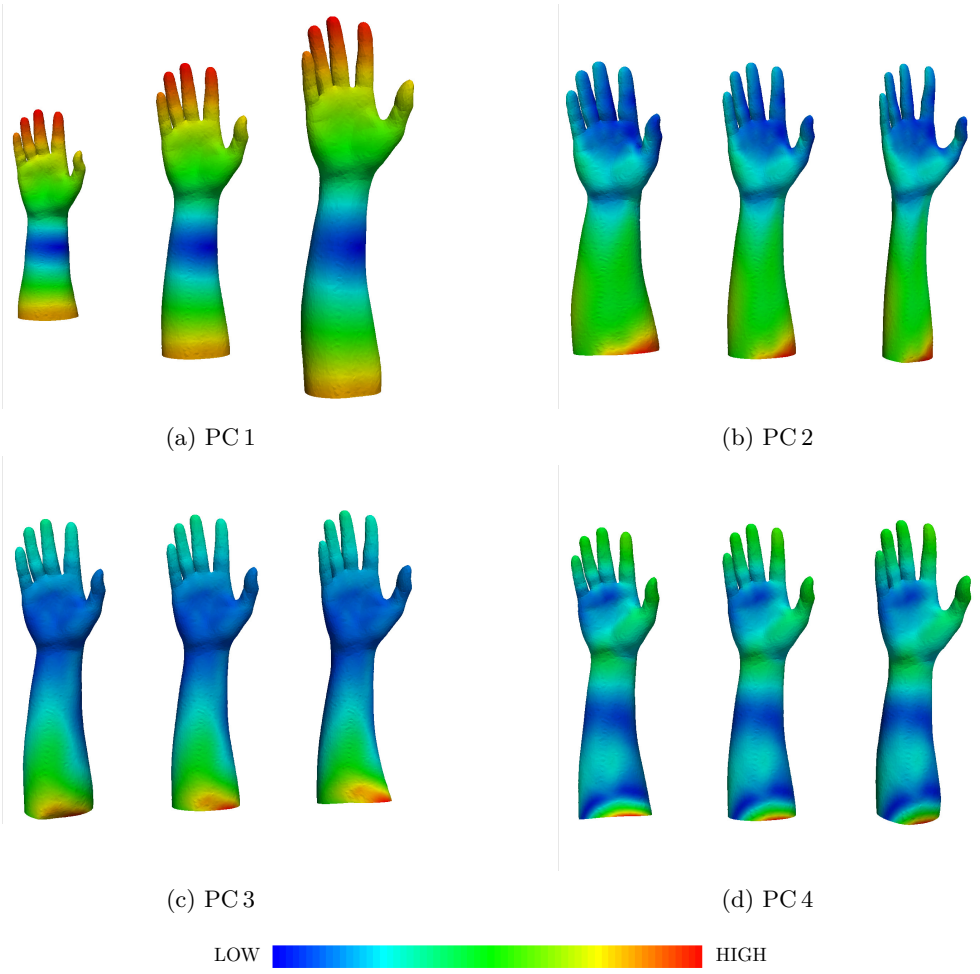


Figure 5.4: First four eigenmodes of the statistical shape model. Color represents the variance $\lambda_i(j)$ for vertex j along the i^{th} PC. For each PC, the shapes are shown which correspond to: $\mu - 3\sigma$, μ and $\mu + 3\sigma$.

5.6 Acknowledgments

This work was supported by the Research Foundation in Flanders (FWO SB) and the VLAIO PLATO-project. The authors would like to thank Vigo nv, More Institute vzw and Orfit Industries nv for their continued contribution to the project.

6

EquiSim: An Open-Source Articulatable Statistical Model of the Equine Distal Limb

Contents

Abstract

Most digital models of the equine distal limb that are available in the community are static and/or subject-specific; hence they have limited applications in veterinary research. In this paper, we present an articulatable model of the entire equine distal limb, based on statistical shape modeling. The model describes the inter-subject variability in bone geometry while maintaining proper jointspace distances to support model articulation towards different poses. Shape variation modes are explained in terms of common biometrics in order to ease model interpretation from a veterinary point of view. The model is publicly available through a graphical user interface (https://github.com/jvhoutte/equisim), in order to facilitate future digitalisation in veterinary research, like computer aided designs, 3D printing of bone implants, bone fracture risk assessment through finite element methods (FEM) and data registration and segmentation problems for clinical practices.

The work in this chapter has been published as:

J. Van Houtte, Vandenberghe, F., Zheng, G., Huysmans, T., and Sijbers, J., "EquiSim: An open-source articulatable statistical model of the equine distal limb", *Frontiers in Veterinary Science*, vol. 8, no. 75, 2021.

6.1 Introduction

Digital three-dimensional (3D) anatomical models have become an important aspect in the digitalisation of veterinary research and medical practices [**?**]. Being acquired by computed tomography (CT) imaging or magnetic resonance imaging (MRI) [**?**] or by 3D optical scanning [**?**], those subject-specific models find their way into finite element analyses (FEA) [**?**], augmented reality guidance during operations [**?**] and training of radiograph segmentation networks [**?**].

A standard procedure in morphological studies of equine distal limb anatomical structures focuses on one-dimensional linear or angular measures, such as: hoof angle, hoof length, medial-lateral width of the phalanges, etc., or two-dimensional measures, such as the joint surface. They are measured from radiographs [**??**], MRI-data [**?**], photographs [**??**] or from in-situ measurements in-vivo [**??**] or post-mortem [**?**].

Another way to study morphology variations is by means of statistical shape models (SSM), which encode the 3D shape variation of the complete bone geometry, rather than reducing the shape to a limited set of discrete measures [**?**]. The benefit of this representation, compared to linear biometrics, is that the statistical shape variability is defined as variation modes of the geometry itself, such that it can be exploited in numerous computer vision applications. In human medical research, these (articulating) SSMs have been widely adopted for training segmentation neural networks on CT-data [**??**]. The models also provide prior shape information for the reconstruction of personalised 3D models from sparse point-data [**?**] or from two-dimensional radiographs [**??**], to facilitate orthopaedic computer assisted surgeries (CAS) or to generate personalised finite element models for mechanical simulations [**??**]. Integrated in deep learning techniques, the models can discriminate between pathological cases based on morphing parameters and thereby outperforms manual subjective classification [**?**]. The inherent geometric information can also be used to study the relationship between shape and biomechanical functions [**?**].

Thanks to additive manufacturing or 3D printing, physical models can efficiently be (re-)produced from these digital models [**?**]. Rapid prototyping has been deployed as didactic material in anatomy classes, to study anatomy besides classical dissection sessions and for training of surgery techniques as an alternative to experimental animals [**??**]. Orthopaedic implant design also benefits from computer aided design (CAD) and 3D printing, as their design can be customised [**??**]. Osteosynthesis plates can be designed specifically to the individual anatomy, prior to fabrication of the plates. CAD thereby omits inter-operative bending of the plates as is the case with off-the-shelf template designs. It has been claimed that customised implant designs which take the shape variability into account improve the clinical outcome [**?**].

Despite the many potential applications, SSMs remain underexplored in veterinary research. Firstly, this is due to lack of availability of large collections of 3D data, from which such model can be built. Most available models are static and subject-specific and are therefore less relevant for CAD. Secondly, there is no one-to-one relation between the variation modes of a SSM and the linear biometrics. This might complicate the interpretation of SSMs and make them less attractive for veterinarians.

In the field of equine veterinary research, we see most potential applications for SSMs to the equine distal limb. The shape of the horse's distal limb bones is an important factor in determining the horse's performance. Because the phalanges and

metacarpal bones distribute the impact forces upon landing on the ground, the shape (and bone mineral density) of the bones affect how efficiently forces are distributed and subsequently determine its risk of fractures [**?**]. It has also been observed that hoof conformation is correlated to movement asymmetry [**?**]. Uneven foot-bearing can eventually lead to biomechanical injuries or lameness, and should be taken into account for corrective shoeing and farriery [**??**].

The aim of this paper is to provide a workflow to generate an articulating SSM of the equine distal limb. Furthermore, the SSM's variation modes are associated with conventional linear biometrics, in order to ease the model interpretation. Unlike earlier SSMs, our model describes the statistical shape variation of the different bones simultaneously in one model. This ensures correct jointspace distances for different model instances and enables articulation of the model towards different poses.

We first outline the methodology to construct an articulating multi-component statistical shape model (aSSM) of the equine distal limb, which is based on the earlier work of Balestra et al [**?**]. Next, we describe the major statistical variation modes, in terms of linear biometrics. In the discussion, we provide directions of future research and potential application areas in the field of veterinary research where the model can be adopted.

6.2 Materials and methods

6.2.1 Data-collection and data-preparation

A random collection of 70 left and right distal front limbs of 35 coldblooded and warmblooded horses and ponies was donated by a commercial abattoir and bulk CT-scanned post-mortem with a Canon Aquilion LB CT system (resolution: $(0.78\times0.78\times0.5)\,\mathrm{mm}^3$, tube current: 200 mA, generator power: 27 kW), from the hoof to the carpus. All legs were unshod at the time of scanning. The hoofs did not undergo prior hoof trimming or cleaning. Right limbs were later mirrored to resemble left limbs.

The acquired CT images were segmented using an open-source graph-cut multi-label segmentation technique [**?**], followed by minor manual corrections. The segmentation label maps were converted to digital geometry surface models by a discrete marching cube algorithm [**?**] and re-meshed to a coarser curvature-adaptive mesh by ACVD [**?**]. Mesh artefacts were eventually resolved by MeshFix [**?**].

6.2.2 Construction of the articulating multi-component statistical shape model

In this section, we describe the proposed methodology to build a compact representation model of the shape variations in our population of $L = 70$ equine distal limb models S_i, $i \in \{0, \ldots, L\}$, with $i = 0$ indicating the reference model which was adopted from earlier work [**?**]. As illustrated in Figure 6.1a, each limb model S_i consists of $M = 10$ components (nine distal limb bones and the hoof capsule), thus: $S_i = \{S_{ij},\ j = 0, \ldots, M-1\}$, where S_{ij} is the j^{th} component of the i^{th} subject with N_{ij} vertices. We denote the homogeneous vertex coordinates of shape S_{ij} by $v_{ij} = \{\boldsymbol{v}_{ijp} \in \mathbb{R}^4,\ p = 0, \ldots, N_{ij}-1\}$. The number of vertices per component j of the reference model are tabulated in Table 6.1 and the resolution of the reference model is visualised in Figure 6.1b.

Bone j	N_{0j}
MC 2	994
MC 3	9159
MC 4	742
PS (lateral)	817
PS (medial)	777

Bone j	N_{0j}
P 1	4226
P 2	2563
P 3	2882
DS	688
hoof capsule	13460

Table 6.1: Number of vertices per component of the reference model.

6.2.2.1 Articulation model

Articulation of the surface model, as illustrated in Figure 6.2 for the different stages of the stance phase, is limited to the major degrees of freedom of the equine distal limb. This includes the extension and flexion around the following three joints: metacarpophalangeal joint (MCP), proximal interphalangeal joint (PIP) and the distal interphalangeal joint (DIP). The articulation model articulates the proximal sesamoid bones and the proximal phalanx as one geometry structure [**?**]. This approximation is justified by their relatively small range of motion and any latent motion which happens in reality will end up as a shape variability in the model. Similarly, the distal sesamoid bone and the hoof capsule are assumed to be rigidly attached to the distal phalanx in the articulation model. Furthermore the three metacarpal bones are rigidly attached to each other. Under these assumptions, the articulation model effectively consists of $N_b = 4$ skeleton bones to transform $M = 10$ surface model components.

The articulation model assigns a local reference frame to each of the four skeleton bones, as depicted in Figure 6.1a. The orthogonal reference frame is defined such that its y-axis aligns with the elongation axis and that its z-axis is perpendicular to the sagittal plane of the bone. The flexion, abduction and internal rotation angle are respectively identified as the three spherical coordinates (α, β, γ) between adjacent reference frames. Their definition is visualised in Figure 6.1c. Note that the flexion rotation axis $\boldsymbol{a}$ of bone b corresponds to the z-axis of its parent bone $p(b)$.

To enable articulation of the distal limb bones themselves, we also define an origin $\boldsymbol{c}$ to each reference frame, which is chosen as the center of a circle, fitted to the joint surface area of the bone in its sagittal plane. The local-to-world transformation $T \in \mathbb{R}^{4\times4}$ brings the local reference frame of a bone, like the one in Figure 6.1c, to its position and orientation in world coordinates. Flexion of a bone b relative to its parent bone $p(b)$ over an angle θ is obtained by the transformation $T_{p(b)}R_z(\theta)T_{p(b)}^{-1}$, where $R_z(\theta) \in \mathbb{R}^{4\times4}$ represents a rotation over the z-axis. It should be noted that the articulation model is a mathematical construction and is not statistically founded by dynamic data.

6.2.2.2 Elastic registration

Initially each subject S_i in the training database is described by its own set of vertices. To statistically describe the shape variations in this database, all shapes must be in semantic correspondence with each other, such that vertices with the same index have the same anatomical location on all training subjects. In order to do so, we elastically deform the reference component coordinates $T_{ij}T_{0j}^{-1}v_{0j}$ towards its corresponding

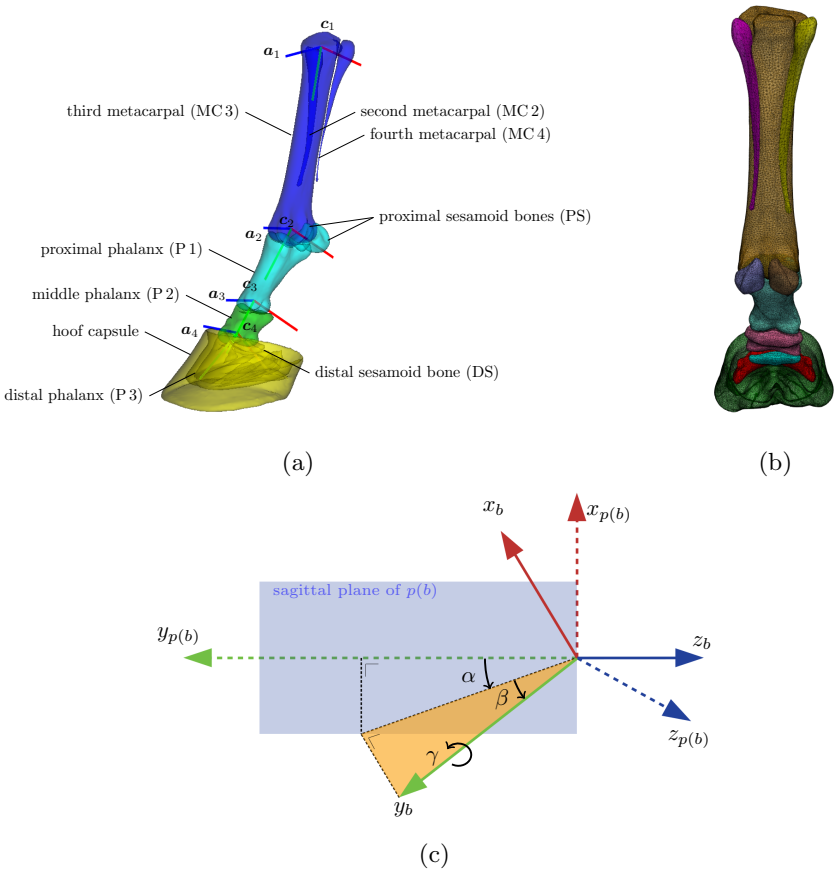


Figure 6.1: (a) The distal limb model consisting of nine bones and the hoof capsule. Bones with similar colors move rigidly under skeleton articulation. Each bone has a local orthogonal reference frame $(\boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b)$ associated to it, which are here represented by respectively the red, green and blue lines. The flexion/extension rotation axis of bone b is denoted by $\boldsymbol{a}_b$ and is positioned at location $\boldsymbol{c}_b$. The flexion/extension around axes $\boldsymbol{a}_2$, $\boldsymbol{a}_3$ and $\boldsymbol{a}_4$ are the major degrees of freedom of the articulation model. (b) Model with overlayed wireframe, indicating the resolution of the surface model. (c) Definition of the spherical angles (α, β, γ) between a bone's reference frame $(\boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b)$, and its adjacent parent bone's reference frame $(\boldsymbol{x}_{p(b)}, \boldsymbol{y}_{p(b)}, \boldsymbol{z}_{p(b)})$. The extension/flexion angle α between two adjacent bones is measured inside the sagittal plane of the parent bone. The corresponding rotation axis $\boldsymbol{a}_b$ coincides with the z-axis of the parent bone $\boldsymbol{z}_{p(b)}$. The abduction/adduction angle β is measured perpendicular to the sagittal plane of the parent bone. The associated rotation axis is $\boldsymbol{y}_b \times \boldsymbol{z}_{p(b)}$. The internal rotation γ happens around the bone axis $\boldsymbol{y}_b$ itself.

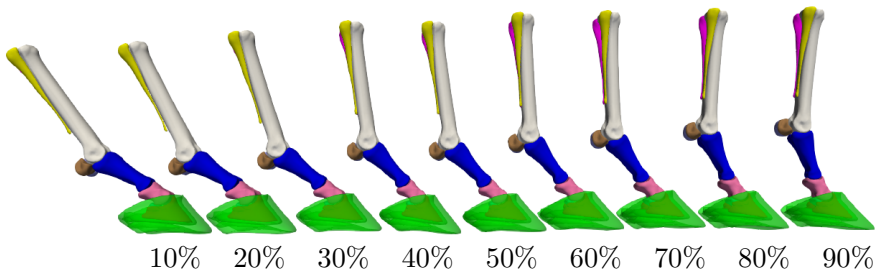


Figure 6.2: Distal limb model's articulation throughout the stance phase. The percentages indicate the completion level of the stance phase. Extension/flexion angles for the MCP, PIP and DIP joints were obtained from the literature [**??**].

training subject component S_{ij} and replace the training subject by the registration result, without change of notation, such that each training shape is now described by the same semantic-meaningful mesh [**?**].

In order to reduce a possible bias towards the chosen reference model, we repeat the elastic registration with the mean model as reference. Note that registered shapes are still in their original position and orientation. Replacing the subject by its registered result introduces a geometric error of how well the original surface is approximated by its registered surface. As illustrated in Figure 6.3, this geometric error is highly position dependent, but overall negligible. The average unsigned geometric error over the entire model equals (0.182 ± 0.002) mm.

6.2.2.3 Scale and pose normalisation

As we are interested in the intrinsic shape variability, we want to normalise all training subjects for their global scale. All models were scale normalised based on the length of the third metacarpal of the reference model.

Secondly, training subjects were originally scanned on their side in different unloaded poses, which causes unwanted pose variations in the dataset. The pose normalisation of bone b involves finding the optimal flexion angle θ^*, in a least-square sense, which matches the bone's geometry with the corresponding bone of the reference model in the local reference frame of its parent bone $p(b)$:

$$\theta^* = \arg\min_{\theta} \|T_{0j}^{-1} v_{0j} - R_z(\theta) T_{ij}^{-1} v_{ij}\|^2, \tag{6.1}$$

where the one-to-one correspondences from the previous step are exploited for the geometry matching. Note that we only optimise for the extension/flexion angle and not for abduction, adduction and internal rotation, which are considered as remnant posture in the shape analysis.

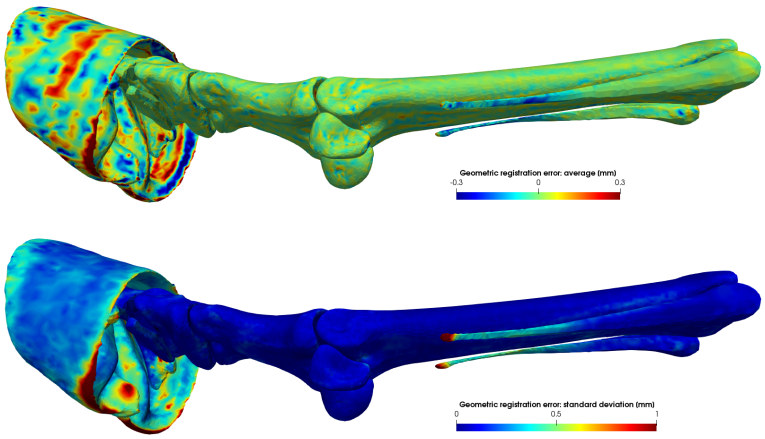


Figure 6.3: Average signed geometric error of the elastic surface registration to $N = 70$ subjects. Vertices of the registered reference model that lie inside or outside the target model have a negative or positive distance, respectively. The average error is a measure of the registration accuracy, while its variance is a measure of the precision of the registration.

6.2.2.4 PCA-based statistical shape modeling

Assuming a database of L registered pose- and scale-normalised shapes S_i, we can define each shape by its shape vector $\boldsymbol{s}_i \in \mathbb{R}^{3F}$, which is a concatenation of its $F = \sum_{j=0}^{M-1} N_{0j}$ coordinates. The shape vectors of the L subjects are ordered as columns in a data-matrix $X \in \mathbb{R}^{3F \times L}$.

The goal of principal component analysis (PCA) is to find an orthogonal transformation which transforms the high-dimensional shape vectors to a low-dimensional set of linearly uncorrelated variables which are called principal components (PC) [**?**]. The PC's can efficiently be calculated by first mean-centering the rows of X and next performing a singular value decomposition (SVD) on the low-dimensional matrix $X^T X$. The matrix X multiplied by the left singular vectors of this decomposition are equal to the principal component vectors $\boldsymbol{u}_i \in \mathbb{R}^{3F}$ of X, after normalising the columns. The singular values of the decomposition are equal to the variances σ_i^2 of those PC's. Figure 6.4 shows the original shapes in a subspace of the $L-1$-dimensional shape space. The PC's are ordered such that the first PC accounts for the largest variation in the dataset, and each succeeding PC has the largest variance possible under the condition that it must be orthogonal to any previous component. Any shape can now be expressed as a linear combination of those PC's, weighted by its standard deviations:

$$\boldsymbol{s}(\boldsymbol{b}) = \bar{\boldsymbol{s}} + \sum_{i=1}^{L-1} b_i \sigma_i \boldsymbol{u}_i \tag{6.2}$$

with $\bar{\boldsymbol{s}} \in \mathbb{R}^{3F}$ the mean shape vector and b_i the contribution of the i^{th} normalised PC $\boldsymbol{u}_i$ to the final shape $\boldsymbol{s}$. In matrix notation, this reads:

$$\boldsymbol{s}(\boldsymbol{b}) = \bar{\boldsymbol{s}} + ED\boldsymbol{b} \tag{6.3}$$

where the columns of matrix $E \in \mathbb{R}^{3F \times L-1}$ contain the normalised eigenvectors $\boldsymbol{u}_i$

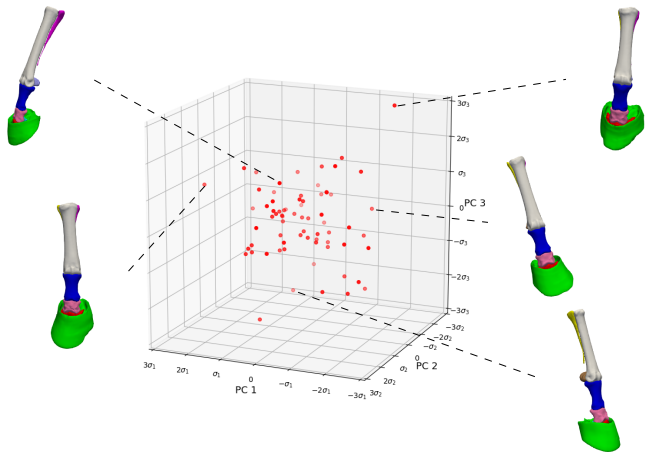


Figure 6.4: Original training shapes shown in a subspace of the $L-1$-dimensional shape space. Only the first three principal component axes are shown. Shapes can be expressed in this shape space in terms of their PC weights $\boldsymbol{b}$.

and $D = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{L-1}) \in \mathbb{R}^{L-1 \times L-1}$. The PC weights $\boldsymbol{b} \in \mathbb{R}^{L-1}$ allow to generate new shape instances from the SSM, different from the training data. Given a new shape $\boldsymbol{s}$ with the same topology as the SSM, the PC weights that approximate this new shape most closely are given by:

$$\boldsymbol{b} = D^{-1}E^{+}(\boldsymbol{s} - \bar{\boldsymbol{s}}), \tag{6.4}$$

with E^{+} the pseudo-inverse of the non-square matrix E.

6.2.2.5 Articulating statistical shape model construction

Applying PCA to the set of registered pose- and scale-normalised shape coordinates $\tilde{v}_{ij} = R_z(\theta^*)T_{ij}^{-1}v_{ij}$, it is important for each shape vector $\boldsymbol{s_i}$ to also contain skeleton information besides the geometry coordinates, because both are intertwined: if the shape changes the underlying skeleton will have to be modified as well in order to maintain proper articulation. The shape vector is therefore a concatenation of the geometry coordinates of all components of the model and the position $\boldsymbol{c}_b$ and orientation $\boldsymbol{a}_b$ of the rotation axes of the bones in the skeleton model, defined in Figure 6.1a:

$$\boldsymbol{s}_i = [\underbrace{\tilde{v}_{i,0}\, \tilde{v}_{i,1}\, \ldots\, \tilde{v}_{i,M-1}}_{\text{geometry coordinates}}\ \underbrace{\log_{\boldsymbol{\mu}_1} \boldsymbol{a}_1^i\ \log_{\boldsymbol{\mu}_2} \boldsymbol{a}_2^i\ \ldots\ \log_{\boldsymbol{\mu}_{N_b}} \boldsymbol{a}_{N_b}^i}_{\text{axis orientation}}\ \underbrace{\boldsymbol{c}_1^i\, \boldsymbol{c}_2^i\, \ldots\, \boldsymbol{c}_{N_b}^i}_{\text{axis position}}]^T \in \mathbb{R}^{3F}, \tag{6.5}$$

with $F = \sum_{j=0}^{M-1} N_{0j} + 2N_b$. Note that we take a logarithmic map of the axis orientation vectors $\boldsymbol{a}_b$ around their intrinsic means $\boldsymbol{\mu}_b$, because PCA assumes that the data is normally distributed in an euclidean (high-dimensional) space. However, the rotation vectors are constrained to lie on the unit sphere S^2, meaning that their length is always one and they can only vary in their orientation. Hence, the vectors lie on a curved manifold and one needs to adopt principal geodesic analysis (PGA) instead of PCA to describe the data variability [**?**]. In short, it applies PCA on the tangent space of S^2, and one needs to use the logarithmic and exponential map around

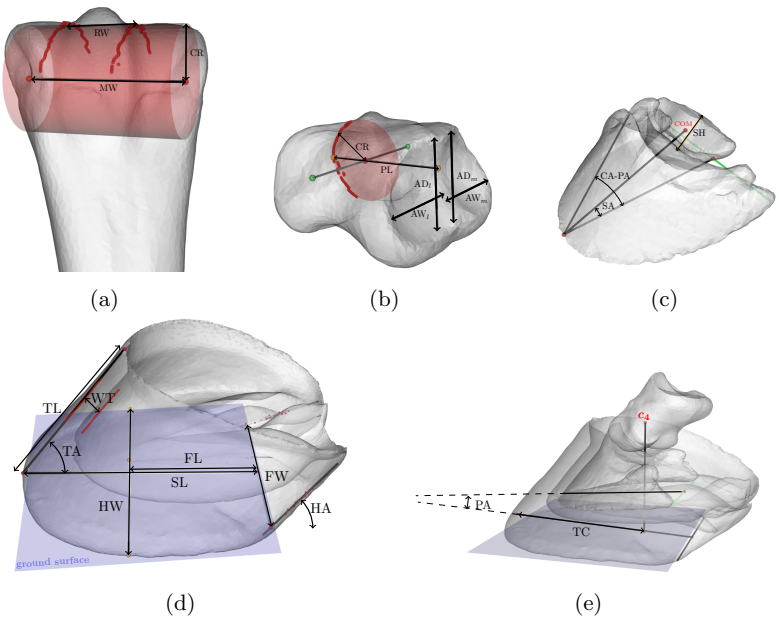


Figure 6.5: Biometrics computed on 3D models. Abbreviations of the metrics are explained in Table 6.2.

the intrinsic mean to move back and forth between S^2 and the euclidean tangent space.

The reconstruction of a component j from the articulating SSM is obtained via:

$$\boldsymbol{s}(\boldsymbol{b}, \theta_j) = (T(\boldsymbol{b}) \circ R_z(\theta_j))\boldsymbol{s}(\boldsymbol{b}), \tag{6.6}$$

where $\boldsymbol{s}(\boldsymbol{b})$ can be calculated from eq.(6.3). Note that the transformation T is also dependent on $\boldsymbol{b}$ as it depends on the rotation axis and center of the bones.

6.2.3 Biometrics

Biometrics are linear or angular measures which characterise a component's shape and can be expressed in terms of the variation modes of the SSM. We consider biometrics for the hoof capsule, the third phalanx [?????], the third metacarpal, the first and second phalanx [?] and the distal sesamoid bone. The definitions of the selected biometrics are tabulated in Table 6.2. The metrics were automatically calculated on the three-dimensional geometry models, instead of on two-dimensional images as is often done in the literature. Figure 6.5 shows the biometrics indicated on the 3D models.

6.2.3.1 Correlation between biometrics and PC modes

In order to change the shape (i.e. changing the PC weights) as a function of the biometric, we applied a multivariate linear regression between the set of PC weights $\boldsymbol{b}$

Bone	Abbr.	Biometrics	Description
hoof capsule	FL	Frog length	Perpendicular distance between the frog apex and the palmar hoof line.
	FW	Frog width	Distance between medial and lateral heel buttress.
	HA	Heel angle	Angle between the hoof wall at the heel and the ground surface.
	HW	Hoof width	Distance between the frog apex and outer capsule wall, measured perpendicular to the sagittal plane.
	SL	Support length	Distance between toe and the palmar hoof line along the ground surface.
	TA	Toe angle	Angle between dorsal hoof wall and the ground surface.
	TL	Toe length	Distance between toe and top of capsule, along the dorsal hoof wall.
	UR	Underrun	The heel angle (HA) minus toe angle (TA).
	WT	Wall thickness	Average thickness of the hoof capsule in the cross-section of the capsule.
P 3	CA	Coffin angle	Angle between dorsal aspect of P 3 and the ground surface.
	PA	Palmar angle	Angle between palmar aspect of P 3 and the ground surface.
	CD	Capsule deviation	Coffin angle (CA) minus toe angle (TA).
	TS	Toe to heel support	Percentage of the hoof's support length (SL) which is ahead of the center of articulation $\boldsymbol{c_4}$ of P 3, ie: TS=TC/SL.
P 1, P 2	CR	Joint curvature radius	Radius of a circle fitted to distal joint surface (lateral view).
	AD	Articular surface depth	Depth of the proximal articular surface, averaged between medial (AD_m) and lateral side (AD_l).
	AW	Articular surface width	Width of the proximal articular surface, averaged between medial (AW_m) and lateral side (AW_l).
	PL	Phalanx length	Length, along major axis in sagittal plane
MC 3	CR	Joint curvature radius	Radius of a circle fitted to distal joint surface (lateral view).
	MW	Medio-lateral width	Width of the distal joint, measured along flexion rotation axis.
	RW	Sagittal ridge width	Average width of the sagittal ridge.
DS	SA	Distal sesamoid angle	Angle between the palmar aspect of P 3 and the line connecting the tip of P 3 with the distal sesamoid's center of mass.
	SH	Distal sesamoid height	Distance between distal and proximal border of the distal sesamoid bone in the sagittal plane.

Table 6.2: Definition of the biometrics.

and the biometric value k:

$$\boldsymbol{b}(k) = \begin{bmatrix} \boldsymbol{\alpha} & \boldsymbol{\beta} \end{bmatrix} \begin{bmatrix} 1 \\ k \end{bmatrix}. \tag{6.7}$$

The regression coefficients $\boldsymbol{\alpha}$, $\boldsymbol{\beta} \in \mathbb{R}^{L-1}$ represent the offset and slope, respectively. They are determined from simulated model instances ($N = 1000$), created from the SSM by using eq.(6.6) following its multivariate normal distribution.

Given a biometric value, the linear regression estimates the PC weights, from which the corresponding SSM's instance can be reconstructed by eq.(6.6). Figure 6.6 shows a number of model instances corresponding to the biometric values $\mu - 3\sigma$ and $\mu + 3\sigma$. Obviously, changing one biometric also changes other biometrics as they are all correlated with each other. In order to visualize the correlations between the different biometrics, we show the Pearson's correlation coefficients in Table 6.3.

6.3 Results

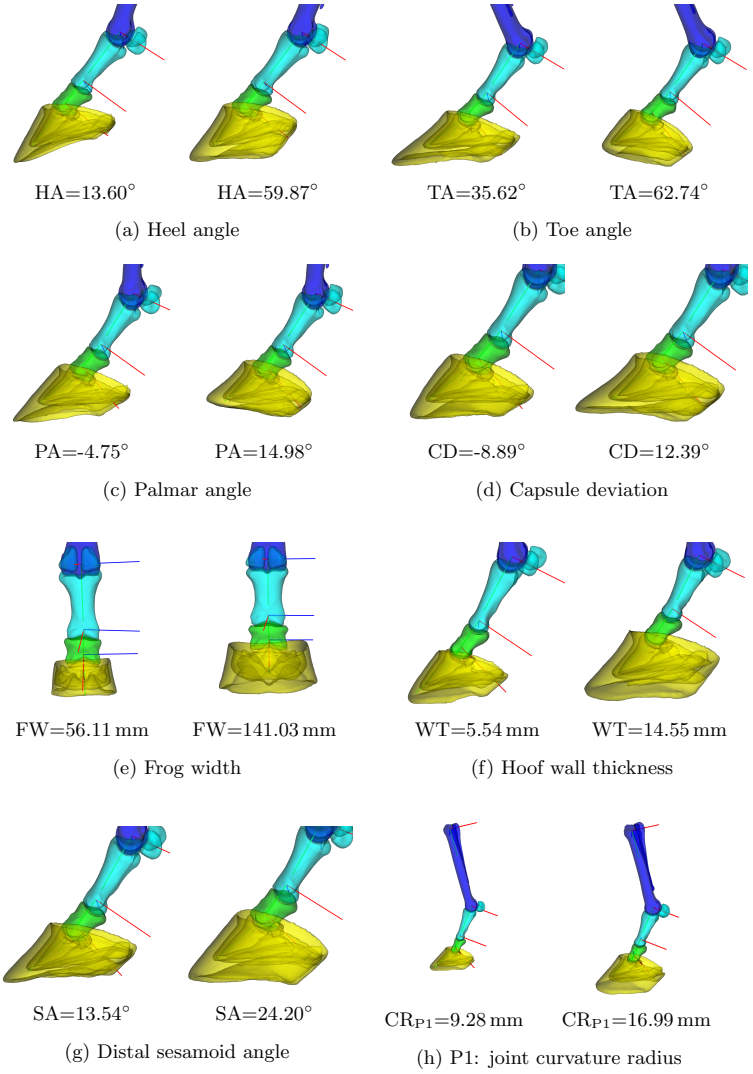
6.3.1 Model performance

The statistical shape model's performance was evaluated in terms of compactness, generalisability and specificity [**??**] and are shown in Figure 6.7 as a function of number of PC modes. The compactness shows the cumulative variance explained by the statistical modes. Figure 6.7a indicates that the first 20 modes describe 95% of the variability in the training dataset. The model's specificity, shown in Figure 6.7b, is the extent to which the model can generate instances of the object that are close to those of the training set. To calculate this measure, random samples of the SSM have been generated by using eq.(6.6), according to its multivariate normal distribution. Each model instance has been compared to its closest training shape in the shape space, in terms of the root mean square error of the distance between corresponding vertices.

The generalisability indicates how well the model can be generalised to new subjects. A leave-one-out test has been performed for calculating this measure. The SSM has been rebuilt for $L - 1$ subjects and eq.(6.4) has been used to fit the obtained model to the subject being left out. Figure 6.7c shows the root mean square error between the subject being left out and the fitted result. We also show the reconstruction error when fitting to the same subject with the complete model, i.e. when the subject to which has been fitted is included in the training data. The discrepancy between both curves is the effect of fitting to unknown subjects. When using the model to fit to new instances one might expect an average geometric error of (2.0 ± 0.3) mm when using the first 20 PC modes, as can be concluded from Figure 6.7c.

6.3.2 Biometrics

The generation of model instances based on a biometric value, as illustrated in Figure 6.6, exploits the multivariate regression relation of section 6.2.3.1. The linear assumption of this regression has been evaluated by recalculating the biometric value $\tilde{k}$ on the model instance $\boldsymbol{s}(\boldsymbol{b}(k), \theta_j)$, created with the biometric value k. The absolute difference between the value k used to generate the model and the recomputed value $\tilde{k}$ on this model results in a confidence interval on k which is reported in Table 6.4



(a) Heel angle (b) Toe angle

(c) Palmar angle (d) Capsule deviation

(e) Frog width (f) Hoof wall thickness

(g) Distal sesamoid angle (h) P1: joint curvature radius

Figure 6.6: Instances of the SSM for different biometric values. For each biometric k the models are shown which correspond to $k = \mu - 3\sigma$ and $k = \mu + 3\sigma$, with μ and σ the average and standard deviation of the biometric in our dataset. Clipped models are shown to draw the reader's attention to the hoof area. In case of the curvature radius (h) one can notice the global scaling of the phalanges and hoof capsule with respect to the third metacarpal.

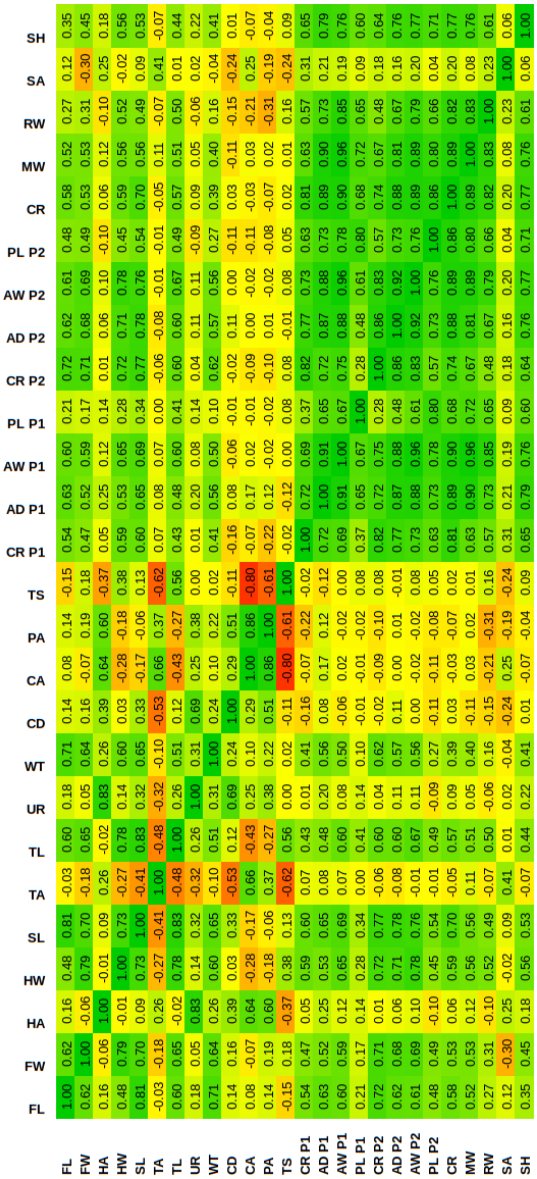
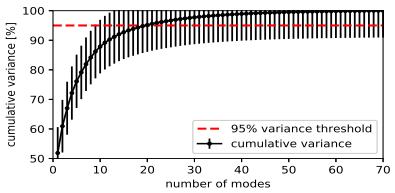
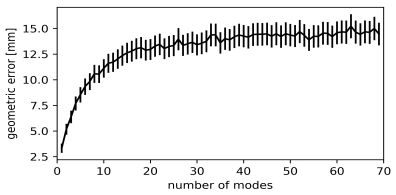


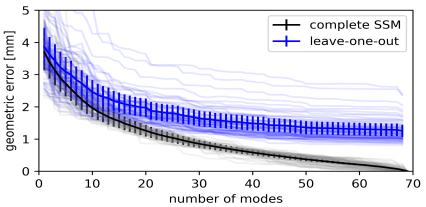
Table 6.3: Autocorrelation table of the biometrics. Correlation values of -1 (red) mean that the corresponding biometrics are anti-correlated, 0 (yellow) means no correlation and +1 (green) means positive correlation. Abbreviations of the metrics are explained in Table 6.2.



(a) Compactness



(b) Specificity



(c) Generalisability & reconstruction error

Figure 6.7: Evaluation metrics of the statistical shape model, as a function of number of PC modes. (a) The compactness shows the cumulative variance explained by the SSM. (b) The specificity indicates how similar randomly generated instances are to the training shapes. (c) Geometric error when fitting the SSM to unseen shapes (generalisability, blue) and to training shapes on which the SSM was built (reconstruction error, black). The faint lines show the geometric error per subject. The solid lines show the average of all subjects.

and gives a measure for the non-linearity of the relation between PC weights and the biometric.

Non-linear effects have dominantly been observed for the angle biometrics, especially for the heel angle and the under-run. Despite the non-linearity, the accuracy of the other angular biometrics is less than two degrees for the full range of $[\mu - 3\sigma, \mu + 3\sigma]$. Most linear biometrics are sub-millimeter accurate. Only the frog width, frog length, capsule deviation and the support length of the hoof capsule have a larger difference between the expected and measured biometric.

6.4 Discussion

In this paper, a workflow has been presented to build an articulating SSM of the left equine distal limb, based on principal component analysis in a pose-normalised coordinate system. As a proof of concept, the workflow has been illustrated on a dataset of 70 cadaver limbs. The resulting model describes morphological variations, while it can be articulated towards any possible pose. We found that 20 modes were sufficient to describe 95% of the population's variability and that it can be registered to new, unseen limbs with a registration accuracy of (2.0 ± 0.3) mm. To ease the morphological interpretation of the resulting statistical modes and to facilitate future research, we have explained the modes in terms of common biometrics and made the model publicly available through a graphical user interface (GUI)[1], shown in Figure 6.8. The source-code of the GUI is developed in C++ on Linux.

[1] https://github.com/jvhoutte/equisim

biometric	min	max
FL [mm]	-1.26	1.41
FW [mm]	-4.46	2.84
HA [°]	-3.34	0.780
HW [mm]	-0.415	-0.224
SL [mm]	-1.27	1.18
TA [°]	-0.502	2.00
TL [mm]	-0.897	0.196
UR [°]	-5.25	1.02
WT [mm]	-0.298	0.156
CD [°]	-1.74	0.882
CA [°]	-0.150	1.37
PA [°]	-0.628	1.04
TS [%]	-0.0231	0.00745

biometric	min	max
CR_{P1} [mm]	-0.0812	0.0225
AD_{P1} [mm]	-0.0925	0.213
AW_{P1} [mm]	-0.0593	0.0726
PL_{P1} [mm]	-0.0201	0.00
CR_{P2} [mm]	-0.00163	0.0131
AD_{P2} [mm]	-0.0929	-0.0189
AW_{P2} [mm]	-0.170	0.151
PL_{P2} [mm]	-0.0220	-0.0153
CR [mm]	-0.000740	0.0430
MW [mm]	-0.150	0.119
RW [mm]	-0.166	0.116
SA [°]	-0.455	0.443
SH [mm]	-0.0457	0.0211

Table 6.4: Confidence intervals for the biometrics.

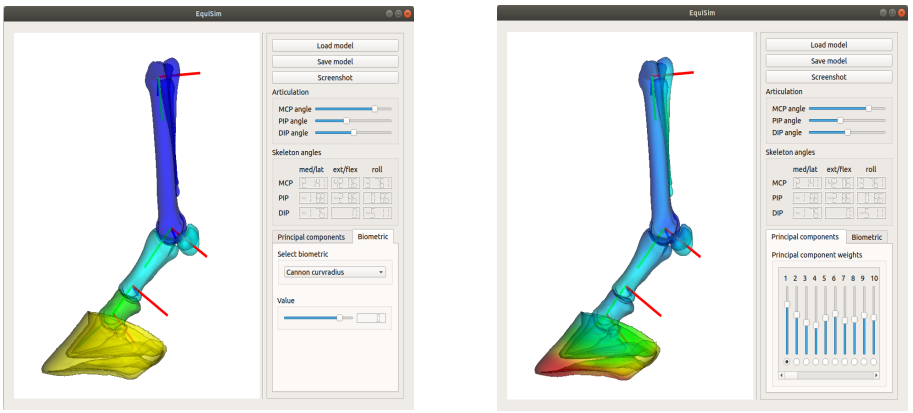
Although the resulting model has been shown to be a compact representation of the population, there are still some limitations which can be a starting point for future research. First, the dataset of equine limbs collected for this proof-of-concept study was ill-controlled, combining different breeds, ages, levels of hoof conditions, etc. This maximized the sample size, but at the same time, made it less relevant for morphological studies. Depending on the target application, it would be beneficial to build a model from a specific dataset.

Secondly, the authors recommend basic hoof trimming and cleaning prior to the data-acquisition. The poor hoof conditions in our collection of limbs caused ambiguities in the data-segmentation and most likely also affected the segmentation accuracy, besides the reported registration accuracy. This subsequently can lead to irrelevant variation modes in the SSM. Similarly, bone ossification between MC 3 and MC 2 or between MC 3 and MC 4 (splints) also posed difficulties in the segmentation of both metacarpals in some cases.

Besides the data-preparation, the model has some more theoretical limitations. The model is not a statistical shape and pose model, in the sense that the pose variations are not statistically described. In this paper, the pose of the model is altered based on a mathematical model, which does not correlate with the shape instance. Changing PC weights does not affect the range of motion. This limitation is due to the difficulty to acquire geometry data in different poses, preferably in-vivo.

Furthermore, the model is a surface model and not a volumetric model. In order to apply our model for finite element analyses [**?**], one still needs to extend the model to a voxelised or tetrahedral model and assign material properties to its cells. The shape variability of our model would enable easy repetitions of the FE analysis for different shape instances. Changing only one biometric allows to study the effect of it on a particular FE result. For kinematical studies, the model can possibly be extended to a musculoskeletal model, by transferring muscle, tendon and ligament attachments from the 3D Horse Anatomy of Biosphera software [**??**].

The main purpose of the model, as presented in this paper, lies in the compact description of the bones statistical variability. This geometric information can be



(a) (b)

Figure 6.8: Screenshot of the graphical user interface "Equisim", which allows the user to interact easily with the model, by either changing the biometric values (a) or by directly changing the PC weights (b).

exploited in CAD of different types of orthopaedic implants, suiting different classes of bone shapes. The ability to create different instances of the SSM also enables the generation of extensive training databases for deep learning applications. Digital or after 3D printing, the model can potentially have educational purposes as well.

6.5 Conclusion

In this paper, we presented a workflow to build an articulating statistical shape model of the equine distal limb, as a way to describe its morphological variations in a compact representation. Three-dimensional shape variations have been related to common one-dimensional biometrics. We thereby bridged the gap between current morphology studies and future digitalisations in veterinary research. Being available through an open-source application, our model can be an added value in veterinary anatomy classes and can potentially support future research in computer-aided designs, finite element analyses and deep learning-based solutions for image processing tasks.

6.6 Acknowledgement

This work was supported by the Research Foundation in Flanders (FWO-SB 1S63918N). The authors would like to thank the equine hospital "Bosdreef" for the CT acquisitions, Denise Vogel for the storage of the cadaver equine limbs and Julie Delhem for her contribution to the CT segmentation corrections.

7

A Deep Learning Approach to Horse Bone Segmentation from Digitally Reconstructed Radiographs

Contents

Abstract

Convolutional neural networks (CNN) are popular for segmentation and classification of bones in radiology. However, their training typically requires a database of thousands of manually segmented experimental images. In many cases, such a large dataset is not readily available in the community. In addition, manual segmentation is often too time intensive and prone to human perception, especially in cases of low image quality. In this paper, we show that a CNN can be accurately trained on the digitally reconstructed radiographs (DRR) of a 3D articulating shape model of the object of interest, bypassing the need for a manually-segmented database. The articulating model ensures a realistic appearance of the bones of interest in the DRR, thereby providing suitable training data for segmentation. As a proof-of-concept, we train a CNN on DRRs with the purpose of segmenting the phalanges of a horse leg from radiographs and show that it outperforms a geodesic active contour segmentation method in this particular case. Our proposed training procedure is effective for articulating objects and the resulting CNN can then be applied to real-data segmentation tasks, if preceded by appropriate augmentation.

The work in this chapter has been published as:

J. Van Houtte, Bazrafkan, S., Vandenberghe, F., Zheng, G., and Sijbers, J., "A Deep Learning Approach to Horse Bone Segmentation from Digitally Reconstructed Radiographs", in *International Conference on Image Processing Theory, Tools, and Applications*, 2019.

7.1 Introduction

The segmentation of radiographs forms the basis of many automated computer-aided analysis pipelines. The segmentation of bones, for example, is of particular interest for diagnosis and monitoring bone disease progression **??**, detecting bone fractures **?** and motion tracking **?**. Segmentation is also a crucial step for 3D bone reconstruction from radiographs, which itself is valuable for pre-operative planning **?** and implant design **?**.

In the last few years, with the emergence of fast and affordable hardware, software, and the availability of large annotated databases, a new machine learning technique known as deep learning (DL) has established itself as a key technology development tool in a variety of fields. From consumer electronics **?** to medical imaging **????** to robotics **???**, one can find the footprints of Deep Neural Networks (DNN) in both regression and classification solutions. In radiology applications, DNNs are adopted for classification, detection and segmentation problems **??**.

Typically, DNN-models rely on a database of ground-truth segmentations for their training. In many situations, however, a database of thousands of examples is not available, or it is too time-consuming to manually segment such large amounts of training data. Furthermore, the region of interest might not always be clearly visible for the expert performing the segmentation, which ultimately increases inter- and intra-operator variability.

To overcome limited training database sizes, synthetic data is often generated to train a DNN **?**, or a limited database is extended by augmenting the data **??**. Yet, to achieve an accurate segmentation result, any synthetic or augmented training data should be representative of actual experimental data. This implies that, in case of segmenting a bone from a skeletal structure, the adjacent bones or structures should be represented as well. To our knowledge, we are unaware of algorithms that create synthetic data of articulated structures for the purposes of training a DNN.

In this paper, we train a convolutional neural network (CNN) on digitally reconstructed radiographs (DRR) of a 3D articulating structure. DRRs are realistically simulated forward X-ray projections of a 3D surface model. The simulation approach ensures a realistic background for the objects to be segmented. The training strategy of the CNN by DRRs bypasses the need for a large manually-segmented database and, by consequence, avoids operator-induced variability in the training data.

7.2 Related work

In this section, we give a brief overview of two state-of-the-art segmentation methods used for bone segmentation from X-ray images. For a general overview of X-ray image segmentation techniques, we refer the reader to **?**.

7.2.1 Deformable models

Active contour models (ACM) are an example of deformable models used for segmentation. ACMs describe the segmentation by an evolving contour which minimises a certain energy function, without imposing prior knowledge on the object to be segmented. This curve can either be explicitly described by a set of predefined points

(snake) or implicitly as the zero-level set of a particular function (level set). The evolution of the curve is typically driven by a shape-regularisation term and a force term which either attracts the contour to an intensity-discontinuity between regions (edge-based) or which optimises the uniformity of a property within a region (region-based). Examples of edge-based and region-based methods include geodesic active contour model (GAC) and active contour model without edge (ACWE), respectively **?**. The region based method relies on the global energy-minimum and is therefore considered as being less sensitive to initialisation and local intensity variations. However, it assumes that the region of interest has a uniform distribution of the image property under consideration. Other deformable models exploit prior shape information of the subject of interest. Point distribution models (PDM) learn the average contour shape along with its statistical variations based on a training database. Such model is iteratively updated by an active shape model (ASM) to find the boundary in an image **?**. ASMs have been extended later to active appearance models (AAM), which also include statistical intensity variations **?**. A variation on AAMs are constrained local models (CLM), which combine a set of appearance models for patches around feature points. Conventional machine learning algorithms, as Random forests, have been used to position these feature points **?**.

7.2.2 Deep learning methods

Deep learning models are made of several processing units including convolutional, fully connected, pooling, and unpooling layers alongside with different normalization and regularization methods such as dropout **?** and batch normalization **?**. The convolutional layers in a DNN apply a sparse mapping to their input. The convolution operation is able to extract the spatial and temporal information based on the signal orientation. In the current work, a feedforward fully convolutional DNN is used to segment the phalanges in a horse leg. A fully convolutional network is a network that only consists of convolution, deconvolution, pooling, and unpooling layers. For more information on the network design see section 7.3.3.

DNNs are playing an important role in the development of semantic segmentation methods. The Fully Convolutional Network (FCN) **?** was first introduced to solve a segmentation problem. This network consists of several convolutional, pooling, and deconvolution layers which resembles an autoencoder scenario due to the fact that the input and output image sizes are the same. But in the FCN network, the output is the segmentation map compared to the autoencoder wherein the target is considered to be the input image.

ParseNet **?** is another fully convolutional end to end DNN wherein the convolutional layers are replaced by a feature extraction module including several normalization steps. The celebrated SegNet architecture **?** provides a high-quality segmentation while it keeps the simplicity in the network structure. This model consists of two fully convolutional networks placed in an encoder-decoder pair shape. The unpooling layers in the decoder take advantage of maximum activations shared by the encoder layers to keep the high-frequency information at the output.

The U-NET **?** is another fully convolutional model and was originally developed for biomedical image segmentation. This network consists of several convolution, pooling, and unpooling layers. The main advantage of this architecture is the existence of skipped connections from the lower layers to deeper layers, which facilitates the passing

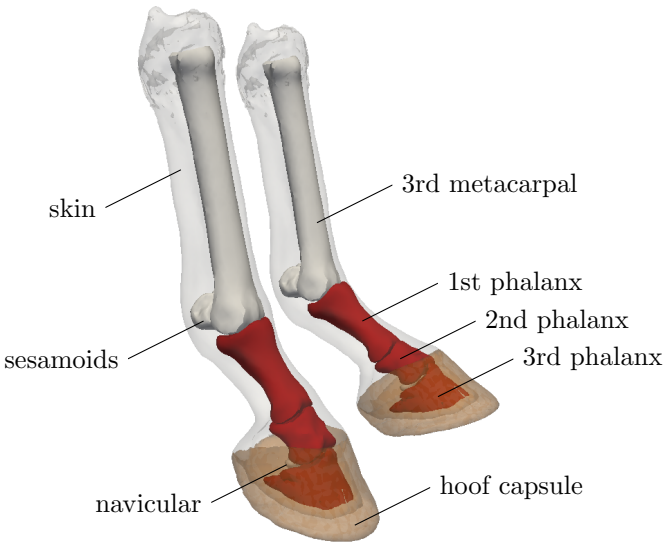


Figure 7.1: Triangular surface model of the horse leg, articulated in two different poses. The full model is used to generate DRRs. The network is only trained to segment the phalanges (red) from the DRRs.

of the high-frequency information from the input toward deeper layers. Another network for binary segmentation presented in **?** merges four different models using a method called SPDNN into a single model. The final architecture resembles a U-net without pooling layers. Instead of pooling layers, larger kernels are used in this network. The authors claim sharper outputs compared to SegNet and U-net in iris segmentation tasks. In the current work, this architecture is used to accomplish the segmentation task due to the simplicity of implementation and high-quality outputs.

7.3 Methodology

Instead of training a CNN on manually segmented experimental data, we propose to train a CNN on DRRs with known ground-truth labels. The creation of this training data consists of two steps: articulating a 3D surface model to a random pose and the simulation of its DRR and its associated label-map. Thereafter, we train a CNN on this artificially created dataset for the purpose of segmenting the phalanges from the DRRs.

7.3.1 Multi-component model

The 3D multi-component model built for the horse leg, shown in Figure 7.1, composes the rigid bones present in the most distal part: the third metacarpal, phalanges, navicular, sesamoids and the hoof capsule. Their triangular surface models were derived from a CT-volume by Panagiotopoulou et al. **?**. They were adopted in this study and equipped with an articulating skeleton, enabling rotation of the phalanges

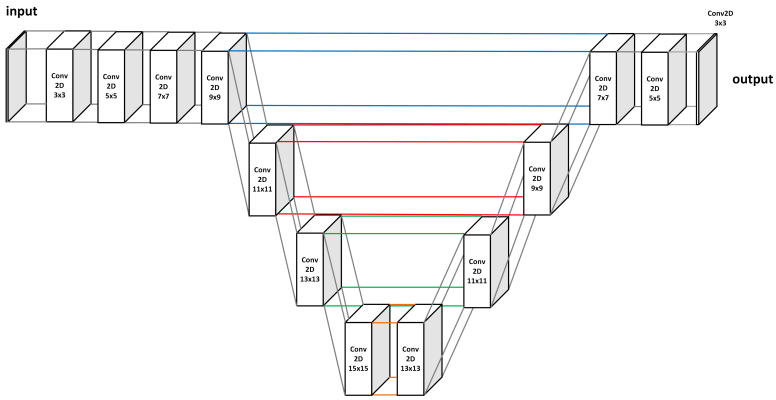


Figure 7.2: The network architecture **?** used to perform the segmentation.

around the interphalangeal hinge joints. This type of joint allows for extension-flexion and only a limited amount of abduction-adduction **?**.

The rotation axis for both types of motion has been calculated for each bone b based on the symmetry plane and the elongation axis of their parent bone $p(b)$. Both rotation axes are perpendicular to each other and have different locations in case of hinge joints. Denoting the local-to-world transformation of bone b in rest and articulated pose by $\hat{C}_b, C_b \in \mathbb{R}^{4\times4}$, respectively, the rest-to-pose transformation of this bone is given by:

$$T_b = C_{p(b)} R(\theta, \phi) \hat{C}_{p(b)}^{-1} \tag{7.1}$$

with $R \in SO(3)$ being the rotation of the bone in the reference system of the parent bone.

The skin layer is also modeled in order to provide a realistic background for the bones in DRRs, DRRs whose creation are explained in the next section. The 3D non-rigid skin deformation due to skeleton articulation is modeled through linear blend skinning **?**. The updated position of vertex i after articulation is given by the weighted average of the original vertex transformed according to the N different bone transformations:

$$\boldsymbol{v_i} = \sum_{b=1}^{N} w_{ib} T_b \boldsymbol{v'_i}, \tag{7.2}$$

where the weights w_{ib} quantify the influence of the bone transformation T_b on the position of vertex i. The skinning weights w_{ib} are calculated by solving a discretized heat-equilibrium differential equation on the skin surface **?**.

7.3.2 Training simulation data

The generation of the DRR is based on mono-energetic ray-casting through the surface model, virtually positioned between a X-ray source point and planar image detector. A ray with source intensity I_0 casted from the source to pixel position $\boldsymbol{p}$ on the detector plane potentially traverses multiple objects on its way. The total length a ray passes through objects with attenuation coefficient μ_i, is denoted by L_i. The intensity at pixel position $\boldsymbol{p}$ is the accumulated result of beam attenuation by all

different objects and is given by:

$$I(\boldsymbol{p}) = I_0 \exp\Big(- \sum_i L_i \mu_i \Big). \tag{7.3}$$

The attenuation coefficient μ is chosen fixed for all bones and chosen larger than the attenuation by the soft tissue.

Along with the DRR, binary ground-truth label-maps are created for each bone with the same dimensions as the DRR. The labelmap for bone j at pixel position $\boldsymbol{p}$ equals zero if $L_j = 0$ and is 1 if $L_j > 0$. Note that each pixel can have multiple labels as different models can potentially overlap in the projection image.
The DRR and label map are resized before being fed to the network training in order to limit the network complexity and computation time.

7.3.3 CNN model and training

The network deployed to perform the segmentation is a fully convolutional network, shown in Figure 7.2. This architecture has already been used in iris segmentation in unconstrained scenarios **?**. The network is designed using a method known as SPDNN **?** which merges smaller networks to design a larger one. The network design workflow is explained in **?**.

The network has 13 layers starting from 3×3 kernels and the size of the kernels increases by getting deeper up to 15×15. From layer 8, the kernel size decreases to 5×5 and at the output layer, a 3×3 kernel is applied. Kernels with size 3×3, 5×5, 7×7, 9×9, 11×11, 13×13, 15×15 are assigned to get 10, 10, 20, 20, 30, 30, and 40 channels, respectively, except the last layer which is a single channel mapping. The ReLU **?** activation function has been used in all the layers except in the last layer which is taking advantage of the sigmoid nonlinearity. There is a batch normalization layer placed after each convolution except the last layer. The loss function used for training is the mean binary cross-entropy given by:

$$L = -\sum_{k=1}^{B_s} \sum_{j=1}^{W} \sum_{i=1}^{H} \frac{t_{ijk}}{B_s W H} \log(o_{ijk}) + (1 - t_{ijk}) \log(1 - o_{ijk}) \tag{7.4}$$

where o and t are the network output and the target respectively, and B_s, W, and H are the batch size, width and height of the images. The parameters updated using the ADAM **?** optimization method with learning rate, β_1, β_2 and ϵ equal to 10^{-5}, 0.9, 0.999 and 10^{-8}, respectively. All the parameters are randomly initialized uniformly between -0.25 and 0.25. The training was done for 1000 epoch on the MXNET **?** framework in python 2.7 on a TitanXp GPU.

7.4 Experiments

The proposed pipeline has been tested on its ability to segment the three phalanges of a horse leg from DRRs. This is in particular a challenging task because these bones are encapsulated by a dense hoof capsule, causing low contrast. Secondly, separation between adjacent bones are usually small.

For the generation of the training dataset we articulated the shape model towards 10.000 different random poses. The poses were constrained within the allowed range

of motions. The global model orientation was confined such that the medial-lateral direction always coincides approximately with the source-detector axis. For each pose, a DRR was simulated, along with the ground-truth label-map.

The CNN was trained and validated on 70% and 20% of the generated database of DRRs, respectively. The remaining 10% was used for testing the network's capability of segmenting the three phalanges. We compared the resulting segmentations with those obtained by geodesic active contour segmentation (GAC), using the open-source implementation of **?**. A visual comparison is shown in Figure 7.3.

The GAC segmentation typically needs to be initialised with a circular region of which the center was annotated by an expert for each image in the test dataset. Besides the initialisation, the outcome is also sensitive to the parameters of the algorithm. The hyperparameters of GAC, explained in **?**, with which we found the best results after fine-tuning were: $\alpha = 650$, $\sigma = 4$, smoothing= 4, threshold=0.5, balloon= 1.

The segmentations by the CNN and the GAC were validated by means of binary metrics, defined in **?**, and are listed in Table 7.1. From that table it is clear that the CNN performs better in segmenting the phalanges according to all six metrics.

The accuracy metric is defined as the ratio of all true results over the total number of pixels. The sensitivity or true positive rate is the probability that the classifier can correctly detect positive pixels and is significantly lower for the GAC than the CNN, indicating that the levelset-classifier often underestimates the actual size the region. Similarly, the specificity or true negative rate measures the probability that the classifier correctly detects background pixels. The high specificity in this case is caused by the large number of background pixels in the images.

The precision or positive predictive value is the probability that a positive outcome of the classifier is true positive. The f1 score is the harmonic average of sensitivity and precision. The Matthew Correlation Coefficient (MCC) is a metric, equal to one for a perfect model, 0 in case of random output and -1 for an inverse segmentation output.

From Table 7.1 we also observe that the metrics of the GAC-method have larger standard deviations than those of the CNN-approach, indicating larger variability in outcome and thus lower predictability for the GAC. This is believed to be caused by the sensitivity of the GAC method to its initialisation and the selected hyper-parameter settings.

7.5 Discussion

The purpose of this paper was to segment the different phalanges of a horse leg from radiographs. Because of low contrast, high noise level and motion blurring, in experimental images we have adopted a CNN-approach, fostered by earlier results **?**.

Because of lack of sufficient training data for this application, we have trained a CNN on DRRs which were simulated from an articulating shape model. We showed that the network is able to correctly segment unseen DRRs and found that it performs better in classifying the different phalanges than GAC-method according to all discussed comparison metrics.

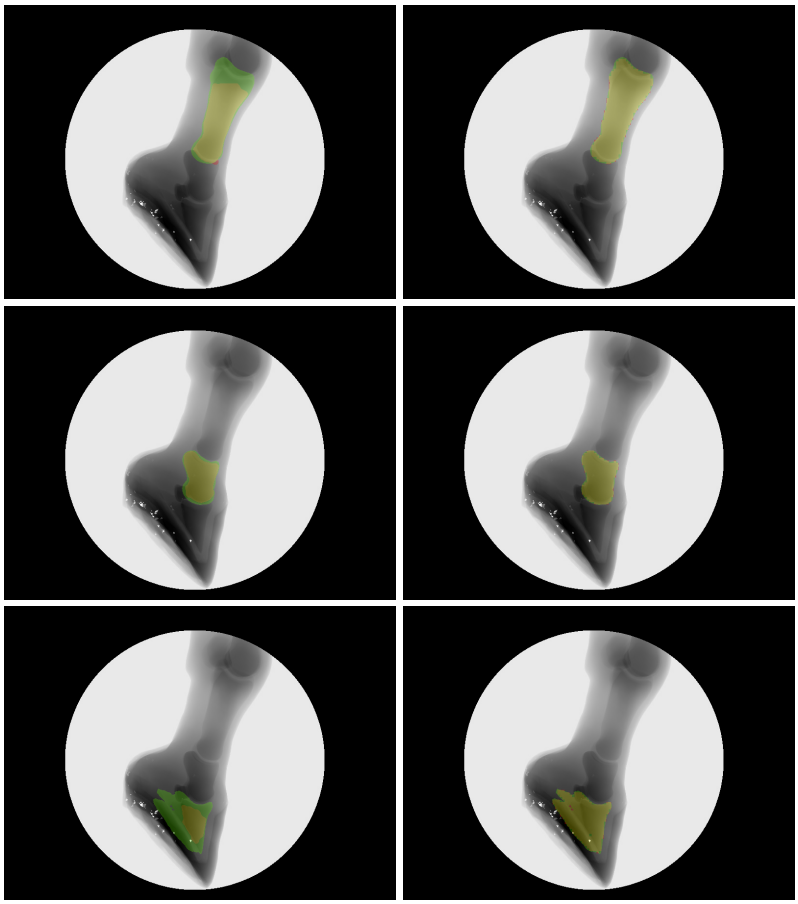


Figure 7.3: Segmentation of the three different phalanges on a DRR test-sample by GAC (left) and the proposed CNN (right). The color indicates the true positive (yellow), false negative (green) and false positive (red) regions. The ground-truth labelmap is thus given by the union of the yellow and green region, while the segmentation is the yellow plus red region.

Table 7.1: Evaluation metrics for the segmentation results.

metric	phalanx 1		phalanx 2	
	CNN	GAC	CNN	GAC
accuracy	**99.911 ± 0.049**	99.20 ± 0.94	**99.910 ± 0.043**	99.26 ± 0.75
sensitivity	**95.5 ± 4.7**	75 ± 17	**93.8 ± 5.1**	64 ± 13
specificity	**99.9954 ±0.0072**	99.78 ± 0.90	**99.9950 ± 0.0073**	99.77 ± 0.77
precision	**99.6 ± 1.7**	92 ± 16	**99.4 ± 4.2**	90 ± 20
f1 score	**97.4± 3.4**	81 ± 13	**96.5± 4.4**	73± 14
MCC	**0.974± 0.031**	0.82 ± 0.13	**0.965± 0.044**	0.75 ± 0.13

metric	phalanx 3	
	CNN	GAC
accuracy	**99.871±0.050**	98.82 ± 0.54
sensitivity	**92.9±3.9**	43 ± 19
specificity	**99.995±0.011**	99.86± 0.49
precision	**99.68±0.76**	92 ± 17
f1 score	**96.1±2.7**	55 ± 17
MCC	**0.961±0.024**	0.60 ± 0.14

The large variation on the metric values of the GAC-method indicates a high unpredictability in outcome. We claim that this is due to the algorithm's sensitivity to the initialization and to the fine-tuning of the parameters. Although ACWE is generally considered as being more robust than GAC, the assumption of uniform intensity within a region did not hold in this specific application and performed less than GAC.

The training strategy of the CNN by DRRs, bypasses the need for a huge manual-segmented database and therefore avoids possible operator-induced variability. Furthermore, the DRR-based training enables the creation of arbitrary large databases and allows to control the amount of pose variability in the database.

However, the current network is not directly applicable on real data because of two limitations in the training process. First of all, the training data does not have the same characteristics as real data, in terms of contrast, intensity, noise, etc. To solve this, the training data should be augmented in order to mimic the real data. Augmentation has been shown to make CNN use-able in real world applications **? ?**, but requires additional knowledge about the image characteristics from a specific acquisition.

Secondly, only one surface model has been used to create the DRRs. As a consequence, the network is not generalised to different individuals. Inter-subject variability can be included in the training data by deforming the DRRs or by adopting a statistical shape model to create the DRRs from. Augmenting the appearance and shape of the training data will be covered in future works.

7.6 Conclusion

We presented a CNN to classify the three phalanges of a horse leg from a DRR-image. The network was trained on DRRs, simulated from an articulating shape model of the horse leg, incorporating the relevant bones and skin surface. Results showed that the proposed CNN outperforms a levelset-based segmentation method. In the future, we plan to augment the training data, to make the network applicable on real experimental data.

Acknowledgment

This work was supported by the Research Foundation in Flanders (FWO-SB 1S63918N).

The authors would like to thank John R. Hutchinson from Royal Veterinary College for sharing the horse CT-data.

We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan Xp GPU used for this research.

Part III

Contributions to DL-based 2D/3D image registration

8

2D/3D Registration with a Statistical Deformation Model Prior Using Deep Learning

Contents

Abstract

Deep learning-based (DL) solutions are increasingly been adopted for 2D/3D registration as they can achieve faster 3D reconstructions from 2D radiographs compared to classical methods. This study proposes a novel semi-supervised DL-network for 2D-3D registration, in which an atlas is registered to two orthogonal radiographs. The deformation of the atlas is composed of an affine transformation and a local deformation constrained by a B-spline-based statistical deformation model. The validaton of the network on digitally reconstructed radiographs from 22 femur CT images shows that the atlas can accurately be registered.

The work in this chapter has been published as:

J. Van Houtte, Gao, X., Sijbers, J., and Zheng, G., "2D/3D Registration with a Statistical Deformation Model Prior Using Deep Learning", in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)* (pp. 1-4). IEEE.

8.1 Introduction

The three-dimensional (3D) reconstruction of bones from two-dimensional (2D) radiographs is crucial in many biomedical engineering domains, such as kinematical studies, pre-operative planning, implant design, and post-operative evaluations **??**. The reconstruction is known to be a degenerate, ill-posed problem, because of the limited number of projections. To resolve the ambiguity of the reconstruction, classical methods tackle the problem as a registration of a 3D atlas to the 2D projections. Statistical models have frequently been adopted to constrain the possible local deformations in a physical way **??**.

Much research in 2D/3D registration has recently turned to deep learning (DL) solutions to achieve real-time 3D reconstructions **?**, being essential for intraoperative guidance and robotic-assisted surgeries **??**. In contrast to the classical methods, current DL approaches often do not register an atlas to the radiographs, but directly decode the 3D image values from the encoded 2D image, ignoring the fundamental degenaracy of the problem **???**.

Being composed of one or two 2D image encoders and a 3D decoder, these networks require a method to bridge between the different dimensionalities of the feature maps, which lacks any connection with the actual physical image generation process. Also, the combination of different projection directions is not physically well founded. The 3D registration to biplanar radiographs is often, by construction, limited to orthogonal radiographs **??**, because of the way in which both directional feature maps are combined.

Cone-beam projections from 3D volumes can actually be simulated by integrating the attenuation along a ray throughout the volume and has previously been integrated in neural networks **??**. Gao et al. generalised the concept of spatial transformers to perspective projections, providing a much simpler and computationally efficient way to simulate cone-beam projections **?**.

In this paper, we propose a semi-supervised end-to-end neural network, which differs from the encoder-decoder architectures proposed in the current literature in that our network estimates a registration field like 3D/3D registration networks **?**. The registration field, being learned from a 3D atlas image and two radiographs, warps the atlas image such that the forward projection of it matches the input radiographs. The deformation field is fully parameterised by an affine transformation and a B-spline-based statistical deformation model (SDM). To the authors' knowledge, this is the first study to present a DL-approach for 2D/3D registration with a B-spline-based SDM.

8.2 Methodology

8.2.1 B-spline-based statistical deformation model

The B-spline-based SDM is constructed from N_s training computed tomography (CT) images, which were registered beforehand to an atlas image $V \in \mathbb{R}^{V_x \times V_y \times V_z}$ by a B-spline-based free-form deformation (FFD). The B-spline coefficients $\boldsymbol{C}$ are defined on a coarse regular lattice of B-spline control points with size $(L+3) \times (M+3) \times (N+3)$. The displacement field that brings the atlas into alignment with each training volume,

is expressed as the 3D B-spline tensor product of 1D cubic B-spline coefficients $\boldsymbol{C}$:

$$\phi(\boldsymbol{C}) = \sum_{r=0}^{3}\sum_{s=0}^{3}\sum_{t=0}^{3} B_r(u)B_s(v)B_t(w)\boldsymbol{C}_{l+r,m+s,n+t}. \tag{8.1}$$

where $B_i(.)$ are B-spline basis functions. The indexes $-1 \leq l \leq (L+1)$, $-1 \leq m \leq (M+1)$, $-1 \leq n \leq (N+1)$ are the indexes of the grid control points, while u, v, w are the relative positions of the image space coordinate in the lattice. As the size of the control point lattice is much smaller than the size of the atlas image, the B-spline FFD gains speed compared to a regular FFD.

The B-spline-based SDM is computed as the singular value decomposition on the set of N_s B-spline coefficient vectors. The SDM expresses any feasible B-spline coefficient vector as a linear combination of the eigenvectors $\boldsymbol{p_k}$ of the decomposition:

$$\boldsymbol{C}(\{\alpha_k\}) = \bar{\boldsymbol{C}} + \sum_{k=1}^{N_m} \alpha_k \sigma_k \boldsymbol{p_k}, \tag{8.2}$$

where σ_k are the associated singular values to the eigenvectors $\boldsymbol{p_k}$. The vector $\bar{\boldsymbol{C}}$ is the average B-spline coefficient vector. The principal component (PC) weights $\{\alpha_k\}$ will act as the model parameters and $N_m \leq N_s - 1$ is the number of selected modes in the model. Each instance $\boldsymbol{C}(\{\alpha_k\})$ determines a forward FFD from the atlas to a floating image by (8.1).

8.2.2 Pseudo-inversion

The forward FFD of (8.1) can be used to warp a floating image backwards to the atlas image domain. For 2D/3D registration, however, this 3D floating image is unknown, and one needs to warp the atlas image backwards by the inverse of (8.1), which is computationally expensive to compute in case of a regular FFD. We therefore apply the pseudo-inversion algorithm on the B-spline coefficients themselves **?**. First, the forward displacement corresponding to the coefficients $\boldsymbol{C}$ is calculated by (8.1). Next, a fixed-point based inversion calculates the inverted displacement on only the control points **?**. Finally, the backward B-spline coefficients $\boldsymbol{C^{bck}}$ on the control points are recursively determined **?**. The 3D B-spline tensor product of (8.1) applied on those backward B-spline coefficients yields the displacement field $\phi(\boldsymbol{C^{bck}})$ that can warp the atlas to the floating image. For more details we refer the reader to **?**.

8.2.3 Projective spatial transform

We use the projective spatial transformer (ProST) from **?** to simulate a 2D perspective projection image $\hat{I} \in \mathbb{R}^{S_x \times S_y}$ from a 3D volume $\hat{V}$. This method defines a fixed canonical grid $G \in \mathbb{R}^{S_x \times S_y \times K}$ of K sampling points, uniformly distributed along each ray connecting the source and each pixel of the 2D detector plane. Given a particular projection geometry, this canonical grid can be transformed by an affine transformation T_{geom} in order to represent the actual projection geometry. The 3D image volume can be interpolated at the transformed grid positions $T_{geom}(G)$. The cone-beam projection is then obtained by integrating along each ray, which is

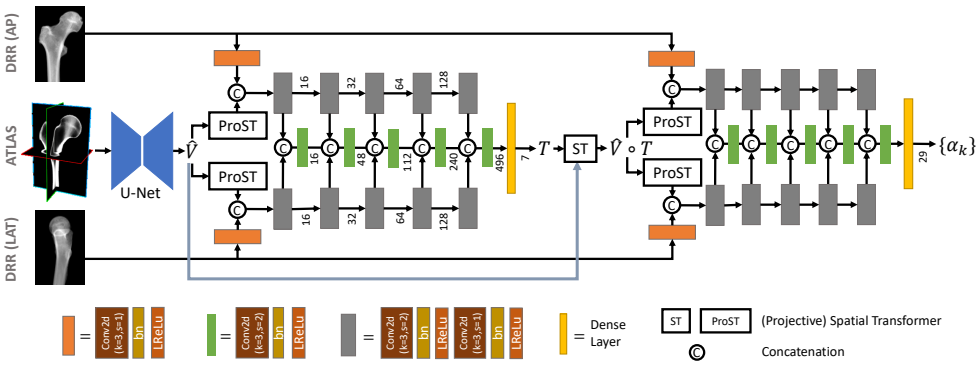


Figure 8.1: Architecture of the end-to-end 2D/3D registration network. The network takes as input two 2D digitally reconstucted radiographs (DRR) and a 3D atlas and estimates a deformation field which is parameterised by 7 affine parameters and 29 PC weights $\{\alpha_k\}$. Both parameter sets are separately regressed by two identical networks. For the first network, we indicate the number of features at each level, which are identical for the second network.

equivalent to a "parallel projection" of the interpolated volume:

$$\hat{I}^{(i,j)} = \sum_{k=1}^{K} (\hat{V} \circ (T_{geom}(G)))^{(i,j,k)}. \tag{8.3}$$

In contrast to **?**, we define two fixed projection angles corresponding to lateral (LAT) and anterior-posterior (AP) projections. Instead of rotating the projection geometry, we keep T_{geom} fixed and apply the affine transformations in the image domain itself to solve the pose problem.

8.2.4 Registration network architecture

The registration network estimates a registration field Ψ that maps the atlas image V (with associated label map S) to the moving image space, such that the forward projection of the warped atlas, $V \circ \Psi$, matches the input radiographs I_i, with $i \in \{\text{AP}, \text{LAT}\}$. The registration field Ψ can be decomposed into an affine transformation T and a local backward B-spline-based deformation field $\phi(\boldsymbol{C^{bck}})$, which is constrained by the SDM. Both components are fully parameterised by respectively 7 affine parameters (rotation, translation and isotropic scaling) and N_m PC weights $\{\alpha_k\}$ of the SDM. The two sets of parameters are separately regressed by two sequential networks, depicted in Figure 8.1.

First a U-net with skip-connections, similar to **?**, learns a 3D volumetric feature map $\hat{V} \in \mathbb{R}^{V_x \times V_y \times V_z \times N_f}$ from the 3D atlas image V, with $N_f = 16$ the number of features. The U-net consists of 4 encoder layers and 6 decoder layers with skip connections in between.

The resulting 3D feature map is projected by a ProST layer, along the AP and lateral direction. Note that the projected feature maps $\hat{I}_i$ still have the same number of features as the volumetric feature map $\hat{V}$. The input radiographs I_i are first convolved such that they have also the same number of features. The ProST output $\hat{I}_i$ and the convolution of the input radiograph I_i are concatenated into a $2N_f$-channel 2D image

and fed to a 2D encoder. Each projection direction i has its own encoder. Each of the five encoder levels consists of a strided convolution, a batch-normalisation layer and a Leaky-Relu activation. Each level reduces the spatial size of the feature map by a factor two and doubles the number of features. At each encoder level, the AP and lateral features (and the preceding combined features) are concatenated and convolved.

The accumulated 2D feature map at the last encoder level is flattened and fed to a dense layer which regresses the seven parameters of the affine transformation T between the floating image and the atlas. The bias and kernel weights of the dense layer are initialised by respectively zero and a narrow normal distribution, such that the initial affine transformation during training is close to identity.

The 3D feature map $\hat{V}$ is warped by the affine transformation T by a spatial transform layer **?**. The transformed 3D features are fed into a similar network as before in order to regress the N_m PC weights $\{\alpha_k\}$, which determine the B-spline coefficients $\boldsymbol{C}$ through (8.2). The pseudo-inversion on $\boldsymbol{C}$ yields the backward B-spline coefficients which determine the backward B-spline-based deformation field $\phi(\boldsymbol{C^{bck}})$ through (8.1). The composition of the affine transformation T and the backward deformation field $\phi(\boldsymbol{C^{bck}})$ is given by: $T \oplus \phi(\boldsymbol{C^{bck}}) = T + \phi(\boldsymbol{C^{bck}}) \circ T$.

8.2.5 Network loss function

The network loss-function, used to evaluate the registration quality during training, consists of a normalised cross-correlation (NCC) between the warped atlas and the ground-truth CT-image V_{gt}, and a Dice loss between the warped atlas label map and the ground-truth label map S_{gt}. Both metrics are evaluated after the affine registration and after the B-spline-based deformation:

$$\begin{aligned}\mathcal{L} = \gamma(&NCC(V_{gt}, V \circ T) + NCC(V_{gt}, V \circ (T \oplus \phi(\boldsymbol{C^{bck}})))) \\ &+ \delta(Dice(S_{gt}, S \circ T) + Dice(S_{gt}, S \circ (T \oplus \phi(\boldsymbol{C^{bck}})))) \\ &+ \zeta \sum_{k=1}^{N_m} \alpha_k^2, \end{aligned} \tag{8.4}$$

with $\gamma = 1.0$, $\delta = 0.1$ and $\zeta = 10^{-6}$ weights to balance the different loss terms. The last term is the Mahalanobis distance and acts as regularisation on the PC weights. It favors instances $\boldsymbol{C}$ of the SDM that are close to the average $\bar{\boldsymbol{C}}$.

8.3 Experiment

8.3.1 Dataset

The training dataset consists of 40 CT-images of naked cadaver femur bones. The validation dataset was acquired separately on different patients and constitutes of 22 CT images, from which the femur bone was masked. The SDM was built on the training dataset. Based on the compactness of the SDM, we have selected the first $N_m = 29$ variation modes from the SDM as they account for up to 99 % of the shape variability in the training dataset. The other modes are regarded as noise and discarded from the set of degrees of freedom optimised by the registration network.

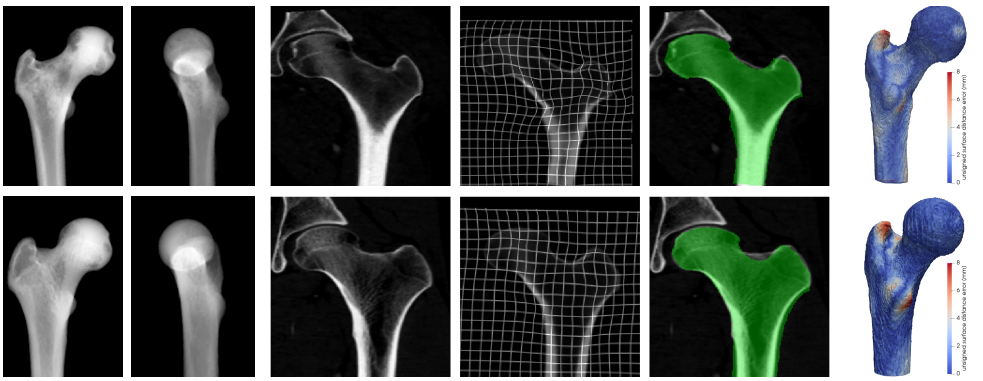


Figure 8.2: Registration of the 3D atlas to orthogonal pairs of DRRs (left columns). The third to fifth column show the same coronal slice of the ground-truth CT-volume, the warped atlas volume together with the deformed grid and the warped label map on top of the ground-truth CT-volume. The last column shows the surface model generated from the deformed atlas segmentation map with the unsigned surface distance error represented by the color map.

The training and validation datasets were augmented off-line by applying random affine transformations on the 3D CT-data, resulting in 1200 and 330 images, respectively. From the transformed CT volumes near AP and lateral digitally reconstructed radiographs (DRR) were simulated with DeepDRR software **?**. The AP and lateral orientations of the femur were defined based on the femoral shaft and neck axis. Pose variations around the perfect AP/lateral view were allowed within a range of 30° internal/external rotation and within a range of 10° extension/flexion and abduction/adduction.

The volume size and voxel spacing of the CT volumes and of the atlas equal $(192 \times 128 \times 192)$ and $(0.66 \times 0.66 \times 1)\,\mathrm{mm}^3$, respectively. The size and pixel spacing of the DRRs equal (141×213) and $(0.9 \times 0.9)\,\mathrm{mm}^2$, respectively.

8.3.2 Results

The entire model, including the pseudo-inversion of the B-spline coefficients and the ProST layer, was implemented in Tensorflow. The network was trained by Adam optimizer for 50 epochs with a learning rate of 10^{-5}, on a NVIDIA Tesla V100 GPU.

The trained model was evaluated on the validation dataset in terms of the Dice metric and the average signed surface distance (ASSD). The average metric values are tabulated in Table 8.1. Figure 8.2 shows two examples of the 2D/3D registration. Ground-truth and estimated surface models were created from the ground-truth label map and the warped atlas label map, respectively. The unsigned distance between those surface models highlight the anatomical features, like the greater and lesser trochanter, as challenging parts to register accurately.

Table 8.1: Average validation metrics

	Dice	ASSD (mm)
Initial	0.515 ± 0.083	8.48 ± 1.49
Affine	0.855 ± 0.038	2.16 ± 0.54
Affine + SDM	0.908 ± 0.018	1.29 ± 0.21

8.4 Discussion

This study presents an end-to-end DL-approach to 2D/3D registration, which differs from the typical encoder-decoder network architectures **?**. Instead of directly decoding the intensity values of a 3D volume without guarantees on feasibility and smoothness of the reconstruction, this model estimates a deformation field that warps an atlas image.

Although we used lateral and AP radiographs in this study, the network is not limited to this particular combination of projections, nor to orthogonal projections. The network can be trained for any combination of projection geometries, as long as the calibration is known beforehand. In the future, we will investigate how re-training the network for each different projection geometry can be avoided.

The network as presented in this study is semi-supervised. The training of the network relies on the auxiliary ground-truth CT-volume and 3D label map associated to the DRR. This type of data is not always available however. Future research could address unsupervised learning schemes for such cases.

As the network is trained on DRRs, the model might not generalise well yet to real experimental radiographs. This will be tackled in future work by augmenting the DRR appearance during training or by including style transfer prior to the network.

9

Deep learning-based 2D/3D registration of an atlas to biplanar X-ray images

Contents

Abstract

The registration of a 3D atlas image to 2D radiographs enables 3D pre-operative planning without the need to acquire costly and high-dose CT-scans. Recently, many deep-learning-based 2D/3D registration methods have been proposed which tackle the problem as a reconstruction by regressing the 3D image immediately from the radiographs, rather than registering an atlas image. Consequently, they are less constrained against unfeasible reconstructions and have no possibility to warp auxiliary data. Finally, they are, by construction, limited to orthogonal projections.

We propose a novel end-to-end trainable 2D/3D registration network that regresses a dense deformation field that warps an atlas image such that the forward projection of the warped atlas matches the input 2D radiographs. We effectively take the projection matrix into account in the regression problem by integrating a projective and inverse projective spatial transform layer into the network.

Comprehensive experiments conducted on simulated DRRs from patient CT images demonstrate the efficacy of the network. Our network yields an average Dice score of 0.94 and an average symmetric surface distance of 0.84 mm on our test dataset. It has experimentally been determined that projection geometries with 80° to 100° projection angle difference result in the highest accuracy.

Our network is able to accurately reconstruct patient-specific CT-images from a pair of near-orthogonal calibrated radiographs by regressing a deformation field that warps an atlas image or any other auxiliary data. Our method is not constrained to orthogonal projections, increasing its applicability in medical practices. It remains a future task to extend the network for uncalibrated radiographs.

The work in this chapter has been published as:

J. Van Houtte, Audenaert, E., Zheng, G., and Sijbers, J., "Deep learning-based 2D/3D registration of an atlas to biplanar X-ray images", *International Journal of Computer Assisted Radiology and Surgery*, 2022, 1-10.

9.1 Introduction

Radiography or X-ray imaging is the most common imaging procedure for many orthopaedic interventions thanks to its ability to visualise internal structures with a relatively low radiation dose and low acquisition cost. Apart from diagnosis, it is a valuable imaging technique for intraoperative guidance and post-operative evaluation. It also plays a crucial role in pre-operative surgical planning and the selection of the right implants. In case of total hip arthroplasty surgeries, for example, it has been shown that the proper positioning and orientation of the acetabular component largely determines the functional outcome of the implant **??**. Thereby, it is essential that parameters such as the centre of rotation of the hip joint, leg length, and hip offset remain preserved after the surgery and thus are correctly assessed on the radiographs.

Although many surgical planning tools rely on two-dimensional (2D) radiographs, their clinical interpretation can be hampered by overlapping structures and magnification effects. The assessment from radiographs can also be influenced by the patient's positioning. To avoid the difficulties associated with 2D projections, three-dimensional (3D) computed tomography (CT) images are preferred for surgical planning because they are less ambiguous **?**. They also allow to study the cortical and cancellous bone, in addition to the outer bone surface **?**. CT-based planning, however, is associated with higher radiation doses and to far more expensive image acquisitions. Previous research has therefore suggested the reconstruction of a patient-specific 3D model from two or more 2D radiographs by registering a 3D CT atlas image to 2D radiographs, referred to as 2D/3D registration **?**.

9.2 Related work

Recently, deep-learning (DL) methods have been proposed that reconstruct a 3D image from 2D radiographs by means of a neural network that encodes the 2D radiographs into a latent variable which is decoded into a 3D CT volume **????**. Compared to 3D/3D registrations, these networks need to bridge between the different dimensionalities in the encoder and decoder, which can be done by reshaping the 2D feature maps **?** or by treating the feature channel as the third spatial dimension **?**. Others exploit the orthogonality between biplanar projections by copying each feature map along a different dimension **?**. The X2CT-GAN network uses two different mechanisms to bridge the dimensionalities **?**. They apply a fully connected layer on the flattened latent variable, before applying a nonlinear activation function and reshaping it into a 3D feature map. For the skip connections, they apply 2D convolutions on the 2D feature maps, which are then copied along the third axis and fed into a 3D convolutional layer.

In this paper, we propose a novel atlas-based 2D/3D registration network that estimates a registration field based on a pair of calibrated radiographs. The main contributions of our proposed method are as follows:

- It follows a registration approach instead of a reconstruction approach, by regressing a deformation field which can be used to warp an atlas or any auxiliary data like segmentation maps. This avoids an additional segmentation step to extract a surface model.

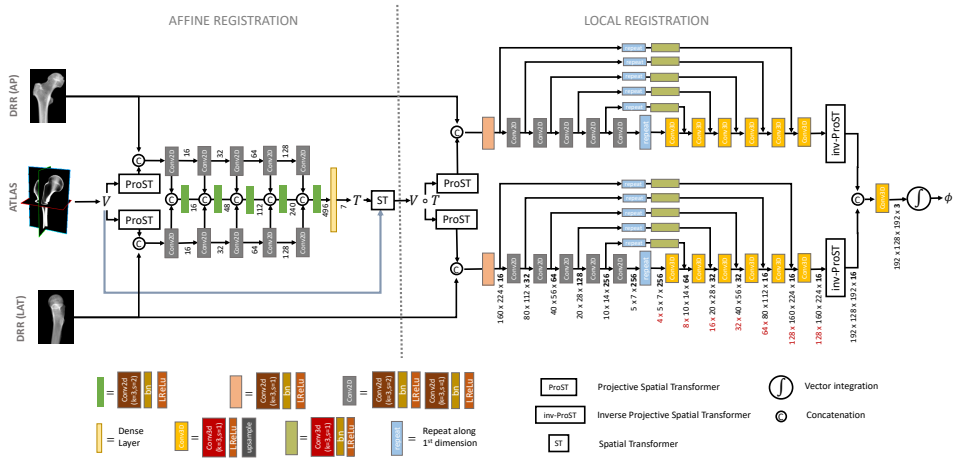


Figure 9.1: Architecture of the 2D/3D registration network consists of an affine and local registration module. The affine module regresses the 7 affine parameters of the transformation T by encoding the anterior-posterior (AP) and lateral (LAT) radiographs. The atlas image V is warped by the regressed transformation before being fed into the local registration module, which regresses the 3D local deformation field ϕ by encoding and decoding the AP and lateral radiographs separately.

- It decomposes the total registration function into an affine and a local part in order to reduce restrictions on the orientation of input data.
- It is not restricted to orthogonal projections, unlike other DL-methods in the literature. To this end, we propose an inv-ProST layer to better combine bi-directional feature maps, as an extension to **?**.
- It is validated on simulated digitally reconstructed radiographs (DRRs) from a large collection of patient CT images, and compared to other registration approaches in the literature **??**.

9.3 Methodology

9.3.1 Registration network architecture

9.3.1.1 Overview of network

The registration network, shown in Figure 9.1, estimates a registration field Ψ that maps the atlas image V (with associated label map S) to the moving image space, such that the forward projection of the warped atlas, $V \circ \Psi$, matches the input radiographs I_i, with $i \in \{\text{AP}, \text{LAT}\}$. The registration field Ψ can be decomposed into an affine transformation T and a local backwards deformation field ϕ. Both transformations are separately regressed by two sequential network modules and composed at the end of the network to yield the total deformation field $\Psi = \phi + T \circ \phi$, which is used to warp the atlas image.

9.3.1.2 Projective spatial transform layer

The projective spatial transformer (ProST), introduced by Gao et al. **?**, simulates a 2D perspective projection $\hat{I} \in \mathbb{R}^{S_x \times S_y}$ from a 3D volume V by sampling this volume at grid locations $G \in \mathbb{R}^{S_x \times S_y \times K}$. The grid consists of K sampling points, uniformly distributed along each ray connecting the X-ray source location to each pixel of the 2D detector. This canonical grid can be transformed by an affine transformation T_{geom} in order to represent the actual projection geometry. This projection geometry transformation T_{geom} is known for calibrated radiographs and serves as input parameter to the network. The 3D volume V can be interpolated at the transformed grid positions $T_{geom}(G)$ to obtain an X-ray beam-aligned image volume $V_{beam} \in \mathbb{R}^{S_x \times S_y \times K}$ in the beam-space:

$$V_{beam} = V \circ (T_{geom}(G)). \tag{9.1}$$

The cone-beam projection is then obtained by integration along each ray, which is equivalent to a "parallel projection" of the interpolated volume:

$$\hat{I}^{(i,j)} = \sum_{k=1}^{K} V_{beam}^{(i,j,k)}. \tag{9.2}$$

9.3.1.3 Affine registration module

The affine registration network consists of two ProST layers which project the 3D atlas image along the AP and lateral direction. The ProST output $\hat{I}_i$ and the input radiograph I_i are concatenated into a 2-channel 2D image and fed into a 2D encoder, corresponding to the ith projection direction. Each of the five encoder levels consists of a strided convolution, a batch-normalisation layer and a leaky rectified linear activation unit (Leaky-ReLU). Each level reduces the spatial size of the feature map by a factor two and doubles the number of features. At each encoder level, the AP and lateral features (and the preceding combined features) are concatenated and convolved.

The accumulated 2D feature map at the last encoder level is flattened and fed into a dense layer which regresses the seven parameters of the affine transformation T between the floating image and the atlas. The bias and kernel weights of the dense layer are initialised by zero and a narrow normal distribution, respectively, such that the initial affine transformation during training is close to identity. A spatial transform layer warps the atlas image V by the affine transformation T **?**, before being fed to the local registration network.

9.3.1.4 Local registration module

The local registration network consists of two separated U-net-shaped networks, each associated with a different projection direction. Each U-net-shaped network is composed of a 2D encoder and 3D decoder and is preceded by a ProST layer that projects the affine transformed atlas image. Each level of the 2D encoder consists of a strided and non-strided 2D convolution. By consequence, each level halves the spatial size and doubles the number of features of the feature maps. After each 2D convolution, a batch normalisation and a Leaky-ReLU activation are applied. The

last 2D feature map is copied $M = 4$ times along the first dimension to obtain a 3D feature map.

The spatial dimensions of the 3D feature map are increased by the 3D decoder, while reducing the number of features as follows: $[64, 32, 32, 16, 16, 16]$. Each decoding step applies a 3D convolution with stride one and a Leaky-ReLU activation, followed by upsampling the feature map by a factor of two. The 3D feature maps are defined in the beam-space, which gives a natural meaning to the above operations. While stacking 2D maps corresponds to increasing the number of sampling points per ray, the upsampling also increases the number of rays.

The network has skip connections between the 2D encoder and 3D decoder at each resolution level of the U-shaped network in order to recover spatial information loss that might have happened during down-sampling. Along each skip connection, the 2D feature maps are copied along the first dimension a number of times, such that its shape corresponds to the 3D decoded feature map's shape. After copying the feature map, the skip connection applies a 3D convolution, a batch normalisation, and a Leaky-ReLU activation to the feature map.

9.3.1.5 inv-ProST

The decoded 3D feature maps are defined in the beam-space and need to be converted to physical space to align them with each other before combining them. Therefore, we apply an "inv-ProST" layer to the 3D feature maps, which samples the feature maps at locations G^{-1}:

$$\hat{V} = V_{beam} \circ (T_{geom}(G^{-1})), \tag{9.3}$$

with G^{-1} the canonical sampling coordinates in the beam space, which are determined by the length of the rays connecting the source location with each voxel and by the intersection point of those rays with the detector plane.

It can be verified that successively applying the ProST of eq.(9.1) and inv-ProST of eq.(9.3) on an image volume V results in approximately the same image V apart from interpolation approximations. Only voxels in the original image that fall outside the cone-beam become zero in the final image.

After the inv-ProST layer, the output tensors of the AP and the lateral network branches can be combined by concatenation, and convolved into a 3-channel tensor which is interpreted as a stationary velocity field. This velocity field is integrated by a "scaling and squaring"-method to obtain a diffeomorphic deformation field ϕ **??**.

9.3.2 Semi-supervised learning

The training of the network is semi-supervised, which means that the training of the network relies on auxiliary data. In our experiment, the segmentation labels of the ground-truth CT volumes were used to mask the image volumes before feeding them into the network, as we are only interested in reconstructing the femur bone from the radiograph images.

The registration quality of the end-to-end network during training and validation is quantified by a loss function that consists of a normalised cross-correlation (NCC) function between the ground-truth image volume V^f and the warped atlas image V

after the affine and local registration. Furthermore, it contains a regularisation term on the smoothness of the local deformation field:

$$\mathcal{L} = -\mathcal{L}_{NCC}(V \circ T, V^f) - \mathcal{L}_{NCC}(V \circ \Psi, V^f) + \delta \mathcal{L}_{smooth}(\phi) \tag{9.4}$$

The hyper-parameter $\delta = 0.01$ balances the contribution between the smoothness term and the image similarity loss. Note that the loss function does not include the label maps anymore, as the images themselves are already masked.

9.4 Experiments

In this section, we evaluate the performance of our network. Section 9.4.1 discusses the generation of the different datasets and provides details on the evaluation and training procedure. Section 9.4.2 presents the registration results for AP and lateral radiographs while comparing with other methods. We also report the sensitivity to inaccurate input parameters and the accuracy for non-orthogonal projections. An ablation study is presented in section 9.4.3.

9.4.1 Experimental settings

9.4.1.1 CT-data preprocessing and augmentation

A total of 315 angio-CT images were acquired and split into a training set of 235 subjects, a validation set of 40 subjects for model selection, and a test set of 40 subjects used to report performance. From each CT-image, the left and right femurs were extracted and rotated to a reference system that aligns the anterior-posterior and lateral views of that femur with the x and y-axis of the image. The femur reference frame of each image was defined based on the neck and shaft axis of the femur. To allow some pose variation around this canonical reference pose, we applied random affine transformations to the image with strict constraints. The randomised angles were allowed within a range of 10° extension/flexion, 10° abduction/adduction and 10° internal/external rotation.
After transforming the images to a pose that is close to that of the reference, the images were cropped around the femoral heads and resized in order to maintain the highest resolution as possible. The left femur images were flipped to resemble right ones. The final CT volumes have a size equal to $(192 \times 128 \times 192)$, and a resolution of $(0.664 \times 0.664 \times 1)\,\text{mm}^3$. Each image has a corresponding segmentation map S, obtained by graph-cut segmentation method followed by manual corrections **?**.

9.4.1.2 Generating DRR

Digitally reconstructed radiographs (DRR) were simulated from the femur-centred CT volumes by DeepDRR software **?**. DRRs were created with an image size of (422×640), and downsampled to (160×224) to fit the network's input size. The source-detector distance and the isocenter distance of the projection geometry were fixed to 1000 mm and 925 mm, respectively. Two different datasets of DRRs were generated:

- A dataset with orthogonal projections. The projection geometry was fixed to provide lateral and AP projections. The acquisition geometry corresponding to this dataset resembles best the experimental settings in the literature.

- A dataset with generalised projection geometries. Projection matrices were parameterised by the left/right anterior oblique (LAO/RAO) angle θ, which was randomly varied between -30° and +30°, around the perfect lateral and AP view. The cranio-caudal angle was set to a constant value of zero degrees. Different combinations of LAO/RAO angles were made for biplanar experiments.

For both datasets, the CT label maps were projected along with the CT images to obtain a 2D labelmap for the DRRs. The DRRs were masked by these labelmaps before feeding them into the network. Note that other structures, in front and behind the femur, are still visible in the masked DRRs.

9.4.1.3 Evaluation metrics

The registration accuracy of the network is evaluated by means of the Dice score and the Jacard coefficient, which measure the overlap between the warped atlas label map and the ground-truth label map **?**:

$$Dice(A, B) = 2\frac{|A \cap B|}{|A| + |B|} \tag{9.5}$$

$$Jac(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{9.6}$$

We also report the average symmetric surface distance (ASSD), which measures the average geometric distance between the ground-truth and registered bone surfaces. The similarity between the warped atlas image and the ground-truth image volume is quantified by the structural similarity index (SSIM), which takes the luminance, contrast and structure into account. As our method is a registration method, its ability to estimate the right intensity values of the image volume is limited. It can only warp an atlas with fixed intensity values.

9.4.1.4 Training details

We implemented our network by using the TensorFlow library. The network was trained for 300 epochs on a NVIDIA Tesla A100 graphics card. The model requires 18.7 GB of memory when being trained with a batch size equal to one, and has a computational complexity of 722 GFLOPS. The loss-function was minimised using the Adam optimizer, with the learning rate set to 10^{-5}.

9.4.2 Experimental results

9.4.2.1 Comparison with other methods

This section describes the results of the registration to AP and lateral DRRs, by our proposed network and by two other networks for comparison. The evaluation metrics are listed in Table 9.1. Figure 9.2 illustrates the qualitative performance of the network by some registration examples.

The first comparison method registers a B-spline-based statistical deformation model (SDM) to a pair of radiographs by regressing its principal component weights **?**. This is a deep-learning implementation of the classical method of Yu et al.(2017) **?**. The SDM guarantees plausible shapes and provides smoother deformation fields than our proposed method, as can be seen in Figure 9.2. Nevertheless, it is outperformed by

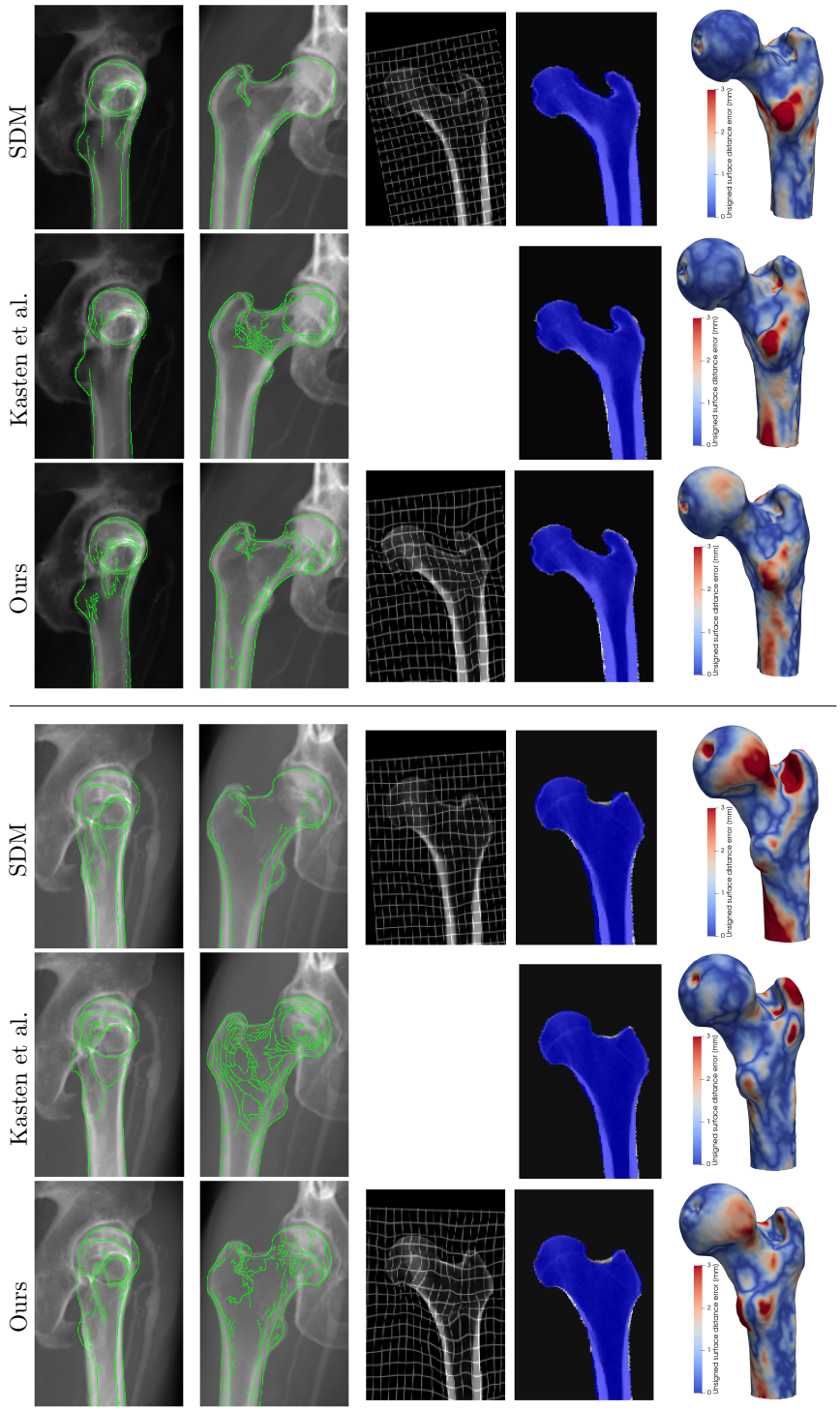


Figure 9.2: Examples of 2D/3D registration based on AP and lateral radiographs by different DL models. The first two columns show the lateral and AP input radiographs overlapped with the contours of the DRRs from the predicted 3D image volume. The third column shows a coronal slice of the warped atlas volume with the deformation grid. The fourth column shows a coronal slice of the predicted segmentation map, overlaid on top of the ground-truth image. The last column shows the geometric reconstruction error between the reconstructed and ground-truth surface model.

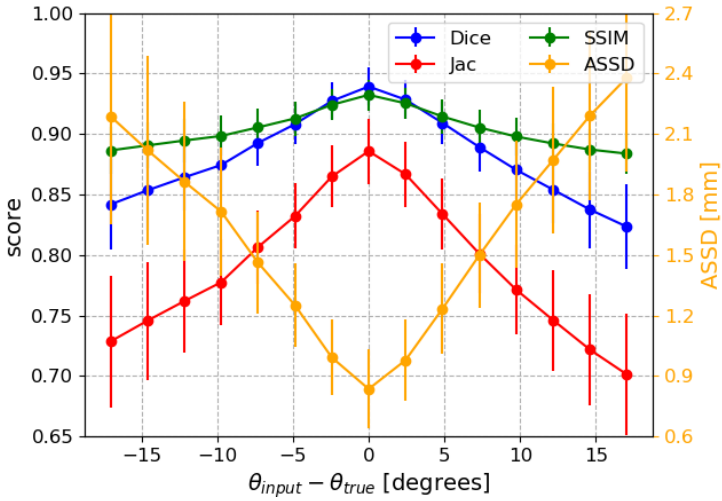


Figure 9.3: Sensitivity of the registration accuracy to inaccurate LAO/RAO projection angle inputs. The true angles θ_{true} in this experiment correspond to perfect AP and lateral views.

our method in terms of registration accuracy ($p = 10^{-30}$), as reported in Table 9.1. This indicates that the constraint on the deformation field by the SDM is too strong to correct for small-scale deformations. The lower SSIM value is due to the different atlas image being used for the SDM-based method. This atlas has an average intensity profile which cancels out more subtle local intensity variations.

The second comparison method is a re-implementation of the work of Kasten et al. **?**, in which the 3D binary labelmap of the femur is immediately regressed from the biplanar radiographs, without deforming an atlas image. This method achieves a larger Dice score than our method ($p = 4 \cdot 10^{-4}$), but lacks information about the internal structures. As it does not regress the 3D intensity values, the problem is considerably simplified.

Figure 9.2 shows a good alignment for our method between the input DRRs and the simulated perspective projections of the registered atlas images, including the cortical bone. The geometric distance error between the estimated and ground-truth surface model highlights the lesser trochanter as a challenging region to register accurately for all methods, while global structures like the femoral neck and shaft are more accurately reconstructed.

9.4.2.2 Sensitivity to inaccurate input

Our network requires calibrated radiographs as input, meaning that the corresponding projection matrix, parameterised by the intrinsic and extrinsic parameters, needs to be known. However, the orientation of an imaging system, like a C-arm system, can never exactly be determined in practice, especially if both projections are taken at different times and the patient moves in between both acquisitions. In this experiment, we study how the uncertainty on the LAO/RAO projection angle affects the registration accuracy for projections which are in reality orthogonal. Figure 9.3

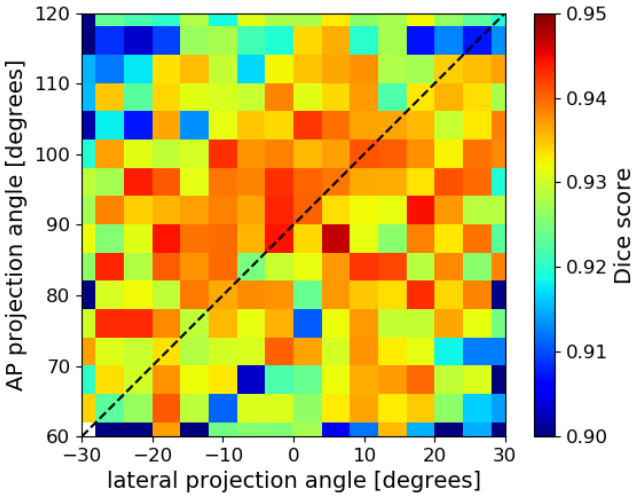


Figure 9.4: Dice scores for different biplanar configurations. Projection angles vary 60 degrees around the perfect AP and lateral angle. The dashed diagonal line shows the configurations with 90 degrees difference between the two projection directions. The bin size is four degrees.

Table 9.1: Registration accuracy of our proposed method and comparison methods **??**.

	Dice	Jac	SSIM	ASSD
SDM **?**	0.921 ± 0.017	0.854 ± 0.028	0.327 ± 0.083	1.16±0.21
Kasten et al. **?**	0.943 ± 0.015	0.892 ± 0.025	–	0.83 ± 0.18
Ours	**0.939±0.016**	**0.886±0.027**	**0.932±0.013**	**0.84±0.20**

shows the evaluation metrics with respect to the difference between the ground-truth and input projection angle. For a discrepancy of five degrees, the average dice score gets reduced from 0.94 to 0.90.

9.4.2.3 Generalised projection geometries

We retrained and evaluated the registration network on the DRR dataset with generalised projection angles. Instead of perfect AP and lateral DRRs, projections were randomly generated in a range of 60° around the AP and lateral views. By training the network on such generalised dataset, the network can be reused for any projection geometry.

The overall average dice score on the generalised validation dataset ($N = 2880$) equals 0.923 ± 0.033. Figure 9.4 shows the median Dice scores for different combinations of LAO/RAO projection angles. The Dice score is maximal for near-orthogonal projection geometries, where the angle between both projection directions is between 80 and 110 degrees. It is interesting to note that projections do not necessarily need to correspond to perfect AP and lateral views.

Table 9.2: Quantitative results for the effectiveness of different network components. The mean and standard deviation (between brackets) of the evaluation metrics are tabulated for the different network variations. The bottom table shows the p-values of a paired t-test between the original network and each variation on the network architecture.

	aff				aff+local			
	Dice	Jac	SSIM	ASSD	Dice	Jac	SSIM	ASSD
Original	**0.860 (0.024)**	**0.755 (0.036)**	**0.893 (0.015)**	**2.04 (0.35)**	**0.939 (0.016)**	**0.886 (0.027)**	**0.932 (0.013)**	**0.84 (0.20)**
2 aff encoders	0.855 (0.023)	0.748 (0.035)	0.891 (0.016)	2.13 (0.34)	0.940 (0.016)	0.888 (0.027)	0.933 (0.013)	0.82 (0.20)
wo skip	0.846 (0.035)	0.734 (0.050)	0.887 (0.016)	2.29 (0.52)	0.937 (0.017)	0.883 (0.029)	0.931 (0.013)	0.86 (0.21)
single 3D dec	0.853 (0.021)	0.744 (0.032)	0.890 (0.017)	2.16 (0.33)	0.930 (0.018)	0.870 (0.030)	0.925 (0.014)	0.97 (0.24)
wo inv-ProST	0.851 (0.026)	0.742 (0.038)	0.889 (0.015)	2.18 (0.36)	0.932 (0.016)	0.873 (0.027)	0.927 (0.013)	0.95 (0.21)
2 aff encoders	10^{-7}	10^{-7}	10^{-7}	10^{-9}	10^{-2}	10^{-2}	10^{-1}	10^{-2}
wo skip	10^{-8}	10^{-8}	10^{-9}	10^{-7}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
single 3D dec	10^{-10}	10^{-10}	10^{-8}	10^{-11}	10^{-21}	10^{-21}	10^{-23}	10^{-22}
wo inv-ProST	10^{-12}	10^{-13}	10^{-13}	10^{-12}	10^{-17}	10^{-17}	10^{-16}	10^{-17}

9.4.3 Ablation study

To study the effectiveness of individual components in our registration network, we re-trained our network, omitting some modules. We used the same dataset as in section 9.4.2 for training, validation, and testing. The evaluation metrics, listed in Table 9.2, are compared to the original results of section 9.4.2 by means of a two-sided paired t-test.

9.4.3.1 Effectiveness of affine network structure

In this experiment, the affine network of section 9.3.1.3 was modified by removing the intermediate concatenations of AP and lateral feature maps. Instead, they were only combined at the end of the affine module, right before regressing the affine parameters. While the affine initialisation is significantly worsened by this, the local registration remains unaffected. It shows that the local registration has a large enough capture range to correct for variations left unseen by the affine initialisation.

9.4.3.2 Effectiveness of skip-connections

Removing the skip connections in the local network significantly reduces the registration accuracy ($p = 10^{-3}$). Secondly, it also increases the training time from 300 to 700 epochs, especially due to the slower training of the affine network. The mismatch in learning rate between the affine and local network can be explained by the vanishing gradient problem. In deep neural networks, the gradient might become very small for the early layers in the network, resulting in a negligible parameter update. The skip connections provide an alternative path to back-propagate the loss-function, which is essential for updating the early network layers.

9.4.3.3 Effectiveness of two separate 3D decoders

Instead of treating the AP and lateral feature maps separately by two distinct encoder-decoder modules, this network variation combines both feature maps at each level

of the 2D encoder, similar to the affine network structure, and only contains one 3D decoder. Skip connections are included between the combined 2D feature maps and 3D decoder. The affine registration module remains the same as depicted in Figure 9.1. The results in Table 9.2 show a highly significant reduction in the affine and local registration accuracy, indicating the preference to decode the 3D feature maps for each projection direction separately.

9.4.3.4 Effectiveness of inv-ProST layer

The inv-ProST layer is responsible for spatially aligning the decoded 3D feature maps into a common coordinate system, before regressing the deformation field. If the inv-ProST layer is left out and the 3D feature maps are directly concatenated instead, the registration accuracy is significantly reduced ($p < 10^{-16}$).

9.5 Discussion

In this work, we presented a DL-model for 2D/3D registration, which substantially differs from other DL-methods in the literature. Instead of directly reconstructing the 3D image volume from a pair of DRRs, like in the model of Kasten et al. **?**, our network estimates a deformation field that can warp an atlas to the floating space. This has the advantage that large deformations and unlikely shapes can be penalised. Secondly, the estimated deformation field can also be used to warp auxiliary data like label maps. Finally, our network is not restricted to perfect AP and lateral projections.

The comprehensive experiments performed on simulated DRRs from patient CT images show the efficacy of our registration method. The network achieves an average Dice score of 0.94 on the test dataset with orthogonal AP and lateral radiographs. While these biplanar views are the standard in musculoskeletal imaging, the acquisition of perfectly orthogonal AP and lateral radiographs is not always achievable in medical practice. Occasionally, instead of horizontal lateral projections, other lateral views, like the frog-leg or Judet view, are sometimes preferred, depending on the underlying disorder **?**. It was experimentally determined that our method still achieves satisfying results for projection geometries deviating from orthogonality by up to $\pm 10°$.

The pair of radiographs that serves as input to our network needs to be calibrated, meaning that the intrinsic and extrinsic parameters of the projection matrix need to be known for both images. The deep network of Gao et al. **?** allow uncalibrated radiographs as input. Their network learns a convex similarity metric with respect to the pose parameters, which is close to the square of geodesic distances in $SE(3)$. In the application phase, this convex similarity function can be optimised over the pose parameters by a conventional gradient descent method. It remains a topic of further research to implement this approach to our network in order to enable uncalibrated radiographs as input and to increase the applicability of the network for medical practices.

Furthermore, the input radiographs to our network need to have the femur masked out. While manually annotating the contours would be a subjective and time-consuming task, automatic methods are proposed in the literature to obtain accurate femoral segmentation maps from radiographs **?**. Selection of the region of interest and

segmentation would also be an important pre-processing step for registration of more complex anatomical structures.

9.6 Conclusion

This paper presents a novel end-to-end 2D/3D-registration network that registers a 3D atlas image to a pair of radiographs. The network regresses a pose similarity transform and a dense deformation field for local shape variations. It effectively accounts for the projection matrix through a projective and inverse-projective spatial transform layer. The experiments show an average Dice score of 0.94 and an average symmetric surface distance of 0.84 mm on the test dataset, which illustrate the effectiveness of our network.

Acknowledgement

GZ is supported by the National Key R&D Program of China via project 2019YFC0120603 and by the Natural Science Foundation of China via project U20A20199. JVH is supported for this research by the Research Foundation in Flanders (FWO-SB 1S63918N). EA is supported by a senior clinical investigator fellowship of the Research Foundation Flanders (FWO). JS and JVH also acknowledge the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" programme.

Conclusion

The registration or alignment of different data with each other is an important pre-processing step in the data-analysis of many biomedical problems. In this manuscript, different registration methods were developed for a versatile of applications through both, classical and deep-learning-based approaches.

Articulating statistical shape modeling

As a registration typically involves the optimisation of many parameters, it is often beneficial to regularise the parameter space to a smaller subspace, by, for example, exploiting the prior knowledge on shape variability through a statistical shape model (SSM). While statistical shape modeling has extensively been studied before for static objects, it was the goal of part II of this thesis to extend the framework to articulating bodies, such as human knees, hands and horse limbs. Compared to a SSM, articulating SSMs introduce additional parameters to control the motion. The true biomechanics, however, is often too complex for this additional articulation model and a trade-off has to be made between the number of articulation and shape parameters. In general, it is better to keep the articulation model as simple as possible and describe remnant motion as effective shape variations.

The framework we have developed in this thesis for building an articulating SSM relies on a mathematical articulation model to normalise the poses of the training subjects, before applying principal component analysis (PCA) on the geometry and pose parameters together. The resulting model can be articulated into different poses for any shape instance, while avoiding model intersections. The skin deformations are, however, not data-driven. Modeling of skin effects, like skin bulging or fat deformation during motion, requires a different unified model to correlate shape and poses **?**. Although the articulation model is not based on dynamic data, the SSM of the bones can possibly be extended to a musculoskeletal model for biomechanical simulations through frameworks as OpenSim **?**.

Articulating SSMs enable a lot of new application areas. Our SSM of the human hand, for example, can potentially be used for automated splint design, based on low-quality 3D scans. While low-quality scans exhibit many artefacts, as holes, sharp edges, etc, a SSM fitted to the 3D scan can give a smooth surface approximation, which is a requirement for product development on top of a surface. Our articulating SSM of the equine distal limb on the other hand is the first of its kind in the veterinary field. In the first place it contains a lot of morphological information, which goes beyond the usual discrete biometrics. This statistical shape information can potentially be used for the design of off-the-shelf horse shoes or orthopaedic implants. The SSM has the ability to generate many instances in different poses which makes it useful

for veterinary training and for the generation of a training dataset for deep-learning models.

Deep learning-based registration

Articulating statistical shape models have also extensively been adopted as prior models in 2D/3D registration by Guoyan et al., among others **??**. The 2D/3D registration aims at aligning a 3D CT-model to one or more 2D radiographs. The usage of a statistical model to bypass the need for a 3D personalised model, reduces the operation cost, time and the radiation dose to the patient.

The goal of Part III of this PhD was to develop a deep-learning-based solution to this problem. Instead of optimising the principal component (PC) weights through a classical optimisation method, a DL-model was developed which regresses the PC weights of a statistical deformation model (SDM) from a pair of radiographs. The drawback of this method is that it involves two separate modeling steps: building a SDM and training the DL-model. Both models require their own independent set of training data.

To avoid the hassle in model training of the first DL-model, a second DL-model has been developed that regresses a 3D deformation field that warps an atlas image, without using a SDM-prior. This approach differs from other 2D/3D registration methods which directly regress the 3D image, without the use of a warping field. The intermediate step of a warping field, gives the possibility to constraint the amount of deformation and to ensure the feasibility of the solution. Secondly, the warping field can be used to warp additional data, like label maps. On the other hand, warping an atlas does not allow for person-specific bone density reconstructions.

Comparing the performances of both networks, it was found that the latter outperforms the SDM-based network. A plausible explanation for this might be that the SDM over-constrains the possible deformations of the atlas. Given the fact that the SDM was built on a relatively small dataset, it is possible that the generalisability of the SDM could still be further improved. Secondly, the different inherent architecture of the networks can play a role in the observed difference, as we did not optimise against the network size in this work. Despite its lower performance and its need for additional training data, the SDM-based network remains an attractive method, because of its simpler architecture and shorter training and inference time.

Generalisability remains a common challenge to, both, statistical shape modeling and deep learning models. Features which were not part of the training database will not be reconstructed, as for example: a broken bone or an implant.

Our developed networks require calibrated radiographs as input, meaning that the intrinsic and extrinsic parameters of the projection need to be known. While the sensitivity of the networks prediction on the extrinsic parameters have been assessed in this thesis, future efforts can generalise the network to uncalibrated radiographs. The current DL-model has been trained for specific intrinsic parameters, meaning that the network would need to be re-trained for different projection geometries. In this respect, the classical registration methods are more easily transferable between different projection geometries.

Another concern regarding generalisability is the performance of the network for

different image appearances, such as image brightness, motion blur, artefacts, etc. As the networks are trained on DRRs, the performance on real radiographs can not be guaranteed, despite the efforts to make the DRRs as realistic as possible. One way to improve the generalisability towards the input appearance is to train a separate GAN network for style transfer.

The presented DL-models were semi-supervised, meaning that the ground-truth 3D image was available during training for each radiograph. This was realised through many simulations, but this is not always possible. Extensions to non-supervised training schemes can be considered in the future.

Despite of the remaining challenges, the solution to the 2D/3D registration can potentially be used in medical situations that currently only rely on radiographs but would benefit from a 3D data representation. This is the case for diagnosis, pre-operative surgical planning, intra-operative guidance and post-operative follow-up.

Curriculum Vitae

Journal papers

- **J. Van Houtte**, Vandenberghe, F., Zheng, G., Huysmans, T., and Sijbers, J., "EquiSim: An open-source articulatable statistical model of the equine distal limb", *Frontiers in Veterinary Science*, vol. 8, no. 75, 2021.
- **J. Van Houtte**, Audenaert, E., Zheng, G., and Sijbers, J., "Deep learning-based 2D/3D registration of an atlas to biplanar X-ray images", *International Journal of Computer Assisted Radiology and Surgery*, 2022, 1-10.

Conference papers

- **J. Van Houtte**, Stanković, K., Booth, B. G., Danckaers, F., Bertrand, V., Verstreken, F., Sijbers, J., and Huysmans, T., "An Articulating Statistical Shape Model of the Human Hand", in *Advances in Human Factors in Simulation and Modeling (AHFE 2018)*, Cham, 2019, vol. 780, pp. 433–445.
- **J. Van Houtte**, Bazrafkan, S., Vandenberghe, F., Zheng, G., and Sijbers, J., "A Deep Learning Approach to Horse Bone Segmentation from Digitally Reconstructed Radiographs", in *International Conference on Image Processing Theory, Tools, and Applications*, 2019.
- F. Danckaers, **Van Houtte, J.**, Booth, B. G., Verstreken, F., and Sijbers, J., "Statistical shape and pose model of the forearm for custom splint design", in *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2021.
- **J. Van Houtte**, Gao, X., Sijbers, J., and Zheng, G., "2D/3D Registration with a Statistical Deformation Model Prior Using Deep Learning", in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)* (pp. 1-4). IEEE.
- X. Gao, **Van Houtte, J.**, Chen, Z., and Zheng, G., "DeepASDM: a Deep Learning Framework for Affine and Deformable Image Registration Incorporating a Statistical Deformation Model", in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)* (pp. 1-4). IEEE.
- **J. Van Houtte**, Sijbers, J., and Zheng, G., "Graphical User Interface for Joint Space Width Assessment by Optical Marker Tracking", in *4th International Conference on Bio-engineering for Smart Technologies*, 2021.

Teaching and supervision

- "Fysica voor biomedisch onderzoek", exercises, first bachelor of Biomedical sciences, University of Antwerp, tutor: prof. dr. Jan Sijbers, from academic year 2016-2017 until 2020-2021.
- "Fysica miv wiskunde", exercises, first bachelor of Biomedical and pharmaceutical sciences, University of Antwerp, tutor: prof. dr. Jan Sijbers, from academic year 2017-2018 until 2020-2021.

Research stays

- Institute for Medical Robotics, Shanghai Jiao Tong University, China, from 8th of July 2019 until 1st of September 2019, supervised by prof. dr. Guoyan Zheng.
- University center of Svalbard, from 29th of october 2021 until 6th of december 2021, supervised by prof. dr. Stein Haaland.