Articulated Statistical Shape Models

Diplomarbeit

zur Erlangung des akademischen Grades
Diplominformatiker

Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Informatik

eingereicht von: Matthias Bindernagel
geboren am: 30. Mai 1982
in: Bernburg/Saale

Gutachter(innen): Prof. Dr.-Ing. Beate Meffert
Dr. Hans Lamecker

eingereicht am: ……

Contents

1 Introduction 1

1.1 Motivation 1

1.2 Objective 3

1.3 Contribution 3

1.4 Thesis Outline 4

2 Related Work 6

2.1 Statistical Shape Models 6

2.2 Modeling Articulation for Image Segmentation 6

2.3 Explicit Joint Models for SSMs 7

2.4 Modeling Natural Joint Posture Variation 8

3 Foundations 9

3.1 Representation of Shape 9

3.2 Representation of Spatial Pose 9

3.3 Prerequisites to Shape Analysis 10

  3.3.1 Correspondence 10

  3.3.2 Alignment 11

3.4 Statistical Shape Models 12

  3.4.1 Point Distribution Model 12

  3.4.2 Varying Pose 13

  3.4.3 Fitting the Model to a Target Anatomy 13

3.5 Segmentation of Medical Image Data 14

  3.5.1 Pose Initialization 14

  3.5.2 Determining Exact Pose and Shape 15

4 A Framework for Articulated Statistical Shape Models 17
4.1 Articulation Model . . . 17
4.1.1 A Compound of Objects with Joints . . . 18
4.1.2 Varying Pose of the Object Compound . . . 20
4.1.3 Determining Individual Object Transformations . . . 20
4.1.4 Fitting Object Poses to a Target Anatomy . . . 22
4.1.5 Determining Partial Derivatives of Object Transformations . . 23
4.1.6 General Specification of Joint Models . . . 25
4.2 Prerequisites to Shape Analysis of Articulated Objects . . . 26
4.2.1 Correspondence . . . 27
4.2.2 Alignment . . . 27
4.3 Point Distribution Model and Articulation Model for ASSMs . . . 28
4.3.1 Point Distribution Model for Articulated Compounds . . . 28
4.3.2 Integration into an ASSM . . . 28
4.3.3 Fitting the Model to a Target Anatomy . . . 29
4.4 Application of ASSMs to Medical Image Segmentation . . . 30
4.4.1 Pose Initialization . . . 30
4.4.2 Determining Exact Pose and Shape . . . 31

5 Implementation 33
5.1 Articulated Statistical Shape Models in Amira . . . 33
5.1.1 Statistical Shape Models . . . 33
5.1.2 Articulated Statistical Shape Models . . . 34
5.1.3 Application of ActiveShapes to Image Segmentation . . . 36
5.2 Fitting Object Poses by Nonlinear Optimization . . . 37

6 Evaluation 40
6.1 Joint Models . . . 40
6.1.1 Model of Spheroidal Joints for the Human Hip . . . 40
6.1.2 Model of the Human Knee . . . 42
6.2 Evaluation Measures . . . 45
6.2.1 Measuring Distance of a Set of Surfaces to Target Surfaces . . 45
6.2.2 Measuring Similarity of Image Segmentations . . . 46
6.2.3 Measuring Invalid Overlap of Segmentations . . . 46

6.3 Boxman 47
6.3.1 Experiment 1: Fitting Pose and Shape by Exploiting Correspondence 48
6.3.2 Experiment 2: Segmentation of Image Data 49
6.4 Segmentation of the Human Lower Limb 52
6.4.1 Preparation of Training Data 52
6.4.2 The Lower Limb ASSM 54
6.4.3 Segmentation Study 54
6.4.4 Results 56

7 Discussion **62**
7.1 Theoretical Performance 62
7.2 Practical Outcome 62
7.3 Improving ASSMs 64
7.4 An Alternative Idea for ASSMs 65

8 Conclusion and Future Work **66**

A Auxiliary Equations and Rationale **69**
A.1 Conversion between Euclidean and Homogeneous Coordinates 69
A.2 Invertibility of Transformations 69
A.3 Basic Transformations and their Elementwise Partial Derivatives . . . 70
A.4 Gradient of the Alignment Objective Function: Auxiliary Calculation 72
A.5 Efficient Calculation of the Alignment Objective Function Value . . . 72
A.6 Efficient Calculation of the Gradient of the Alignment Objective Function 73

B Examples **75**
B.1 Determining Object Transformations 75
B.2 Determining Corresponding Target Positions during Image Segmentation 77

C Additional Evaluation Figures **79**

Bibliography **82**

List of Symbols

α	$\in \mathbb{R}^q$, vector representing all q pose degrees of freedom of an articulated compound, i.e. concatenation of the components of α_k of all joints and the global α_g
α_g	Vector representing the degrees of freedom of the global pose of a compound and consisting of 3 rotational, 3 translational and one optional uniform scaling component
α_k	Vector representing the degrees of freedom of the k-th joint of a compound
b	$\in \mathbb{R}^K$, shape weight vector, i.e. degrees of freedom regarding shape variation
J_k	$\in \mathbb{R}^{4\times4}$, rigid, relative transformation representing the posture of the k-th joint of a compound
K	Number of shape modes
M	Number of objects of a compound
N	Number of vertices of an object
N_i	Number of vertices of the i-th object of a compound
P	$\in \mathbb{R}^{3N\times K}$, matrix of eigenvectors of the covariance matrix
q	$\in \mathbb{N}^+$, number of degrees of freedom of the articulation model, that is sum of the numbers of degrees of freedom of all joints and the *global transformation*
T	$\in \mathbb{R}^{4\times4}$, rigid transformation, optionally including uniform scaling

T_g	$\in \mathbb{R}^{4\times 4}$, rigid transformation, optionally including uniform scaling, representing the pose of the *fixed* object of a compound and hence the pose of the compound
T_i	$\in \mathbb{R}^{4\times 4}$, rigid transformation, optionally including uniform scaling, representing the pose of the i-th object of a compound
u_i	i-th vertex of a reference object in three-dimensional or homogeneous coordinates
u_{ij}	j-th vertex of the i-th reference object in three-dimensional or homogeneous coordinates
V	$\in \mathbb{R}^{3N}$, vector containing the components of all v_i of an object or all v_{ij} of an object compound
$\bar{V}$	$\in \mathbb{R}^{3N}$, vector containing all vertex coordinates of the mean shape of an object or a compound of objects
v_i	i-th vertex of an object in three-dimensional or homogeneous coordinates
v_{ij}	j-th vertex of the i-th object of a compound in three-dimensional or homogeneous coordinates
w_i	Displacement weight of the i-th vertex
w_{ij}	Displacement weight of the j-th vertex of the i-th object

1 Introduction

1.1 Motivation

Statistical shape models (SSMs) capture the shape and its natural variation of an object. An SSM is learned from a set of training shapes which defines the space of allowable shapes. The resulting model can then be deformed within the boundaries of this shape space, as depicted in Figure 1.1.

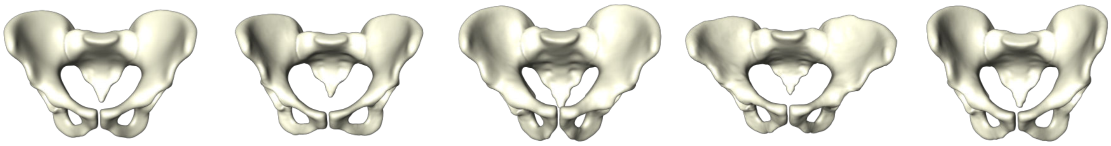
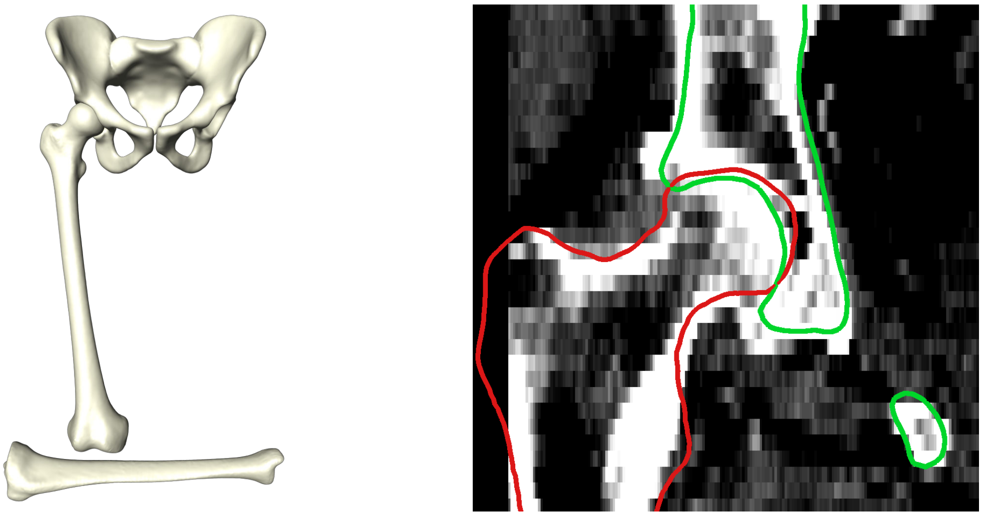


Figure 1.1: SSM of the human pelvis: Mean shape (left) and extremal shapes with respect to the first two modes of variation

In the domain of computer-aided surgery, SSMs are a widely-used tool to reconstruct a patient-specific anatomy from partial or unclear anatomical information. 3D image segmentation denotes the problem of assigning a label to each voxel of a tomographic image and thus identifying it as part of a particular object. Here, SSMs are employed to reliably recognize the complete shape of an object even in presence of distorting image artifacts or in absence of distinct features. Another application of SSMs is the estimation of a patient's healthy anatomy in the presence of pathological changes or when parts of an object have been resected.

These applications benefit from the fact that SSMs represent different, but always reasonable shapes of a class of objects. Given an extensive set of training shapes, it is said that an SSM may approximate most naturally occurring anatomical shapes. These properties make these models a valuable and robust tool when anatomical objects need to be reconstructed approximately.

Concerning the application of SSMs to the anatomy of joints and their involved bones, one must also consider the variation of joint posture. A common idea is not to care about joints at all and hence to employ multiple, independent models of the individual bones. However, there are two major drawbacks with this approach: First, an object's pose is independent of its adjacent object(s). This allows arbitrary object poses that do not resemble natural joint postures, as shown in Figure 1.2a. Second, the shape of two neighbor bones is decoupled, although the adjacent surfaces of contact may correspond with regard to their shape. A failed reconstruction of the closely spaced acetabulum of the pelvis and the femoral head can be seen in Figure 1.2b.



(a) The human lower limb in an anatomically very unnatural state as it may be represented by individual SSMs of pelvis, femur and tibia

(b) Failed hip segmentation by individual SSMs of pelvis (green) and femur (red): The shapes of the acetabulum of the pelvis and the femoral head do not match and thus produce an overlapping segmentation

Figure 1.2: Shortcomings of individual SSMs representing a compound of objects

To eliminate these shortcomings, the individual SSMs may be integrated into one model that considers degrees of freedom that are better suited to such object compounds: Object poses may be coupled with respect to joints' natural degrees of freedom and the shape of all comprised objects may be analyzed and varied together. With [KLZH09], Dagmar Kainmüller has shown that such an articulated SSM (ASSM) of the human hip increases the reconstruction accuracy of a fully-automated CT image segmentation of pelvis and femur. Inspired by that work, the author has implemented an ASSM of the human knee with the intent to interpolate a patient-specific, healthy anatomy in the presence of pathological changes. Details can be found in [BKSLZH11].

The work at hand presents a generalization of the ASSM approach to arbitrary articulated structures comprising multiple objects and joints. The resulting framework relies on explicitly modeled joint representations that map anatomical degrees of freedom of joint motion onto relative, rigid transformations between adjacent objects. This way, ASSMs always assure anatomically valid states of the represented structures. The implementation of such ASSMs is demonstrated with the help of exemplary models. In the end, the performance of the approach is evaluated by segmentation of clinical, tomographic images.

1.2 Objective

The goal of this work is to build a general framework for articulated statistical shape models that consist of multiple, jointed objects. This framework shall be able to cover a broad range of joints and shall allow an easy implementation of models of arbitrary articulated structures that resemble their natural degrees of freedom of articulation. The resulting models shall substitute conventional SSMs as component of a segmentation pipeline which performs the entire, fully automatic segmentation including object detection, SSM adaption and subsequent post-processing. The integration of ASSMs should not require substantial changes to the pipeline itself. The segmentation performance of the new approach will be evaluated in comparison to usual SSMs.

1.3 Contribution

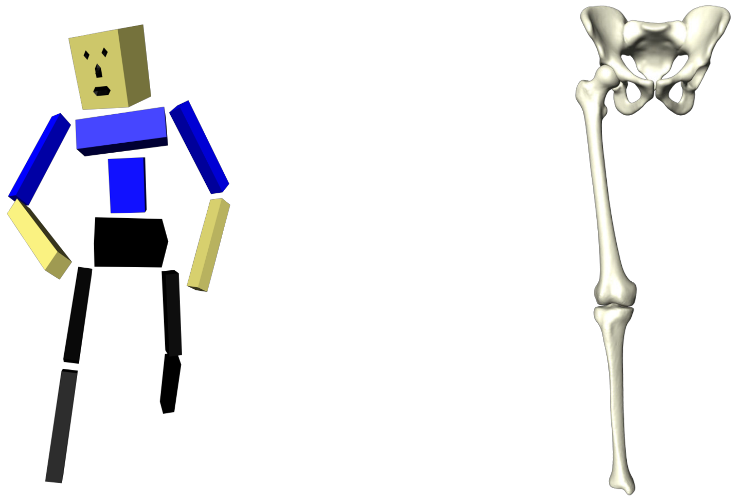
This work presents a general framework for articulated statistical shape models (ASSMs) as an extension of conventional SSMs. These models combine statistically modeled variation of shape and physical variation of pose of individual objects. The relative pose of these objects is restricted by the natural, explicitly modeled degrees of freedom of respective joints.

The ASSM framework supports

- multiple objects as part of an articulated compound,
- a simple interface for easy implementation of particular joint types and
- seamless substitution of conventional SSMs with ASSMs in an established segmentation pipeline.

Its performance is demonstrated with the help of the toy example Boxman depicted in Figure 1.3a. It consists of 12 objects connected by 11 joints.

To evaluate the capability of the approach for medical image segmentation a study with an ASSM of the human lower limb, as shown in Figure 1.3b, is conducted. Segmentations of CT images based thereon are compared to segmentations stemming from conventional SSMs.


(a) Toy example Boxman

(b) Human lower limb

Figure 1.3: Articulated statistical shape models presented in this work

The implementation of these particular ASSMs includes the specification of two joint models: A generic rotational joint and a model of the human knee. An ASSM of the human knee based on the latter model has already been described in [BKSLZH11]. Furthermore, a preview of this work has been presented in [BKRLZ12].

1.4 Thesis Outline

The work at hand is structured as follows. First, the author discusses approaches that cope with articulated structures with regard to segmentation and anatomy reconstruction in Chapter 2. The next chapter then describes the inner mechanisms of statistical shape models for medical image segmentation as they are the foundation of the method proposed here.

In Chapter 4 the methodological approach of articulated statistical shape models is presented. Section 4.1 describes the model of articulation that enables individual positioning of objects so that their relative position to adjacent objects reflects a natural and valid joint posture. The remainder of that chapter deals with the combination of articulation and shape variation to constitute articulated statistical shape models and considers their application to segmentation tasks.

The realization of computational tasks described therein and a particular implementation of the framework are described in Chapter 5. The practical capabilities of the presented approach are evaluated in Chapter 6 with the help of two ASSMs: The artificial Boxman example and the model of the human lower limb (see Figure 1.3).

In Chapter 7 the results of the evaluation and the advantages and drawbacks of the presented approach are discussed. The final chapter then concludes with prospects and ideas that could further improve the ASSM method.

2 Related Work

2.1 Statistical Shape Models

Statistical shape models (SSMs), as presented by [CTCG95] and also known as active shape models, represent the shape and its variation of one type of object, e.g. a particular bone. They constitute a widely-used tool for medical image segmentation, anatomy reconstruction and statistical analysis of shape in general. For example, [LSHD04; SKHLZH08] present an SSM of the human pelvis and embed it in a pipeline for the fully automatic segmentation of medical image data. For a detailed overview of SSMs and related concepts in the context of shape analysis, the reader is referred to [DM98].

Details of the inner mechanisms of SSMs are presented in the next chapter as a necessary basis for the implementation of ASSMs.

2.2 Modeling Articulation for Image Segmentation

There already exist approaches that address the stated problem concerning segmentation of articulated anatomies.

In the work of [HH95] an extension of SSMs is proposed to capture angular shape variation properly. This way some aspects of joint posture variation like bending and pivoting can be captured more accurate by such an extended model.

Conventional SSMs capture variation of shape linearly, that is, model vertices move along straight lines regarding one mode of variation. Non-linear variation must be approximated by combination of multiple modes of variation. However, regarding the collective rotation of many points – which is a common example of joint variation – this approximation lacks accuracy.

To overcome the limitation of linearity, the authors propose models that perform a statistical analysis on vertices either in Euclidean form or in polar coordinates. Shape and joint variation can then be learned from a set of training shapes and are not explicitly distinguished. However, a drawback of this approach is that it requires specifically prepared training data with varying joint posture. Furthermore, a vertex is varied in either angular or linear manner, a combination of both aspects is not realized.

A different approach for the segmentation of the human spine was developed by [KWLFO08]. Since the individual vertebrae are very similar in shape, a major issue with conventional methods is that a single object model targeted at a particular vertebra may drift towards a neighboring one during automatic adjustment.

To avoid such misplacements the adaption is divided into two stages. During the first stage the position of the individual vertebrae is detected by fitting a compound model of the spine. This model consists of rigid vertebrae representations whose poses can be adjusted according to learned, relative transformations. Its purpose is to initialize object positions accurately for the next step. In the second stage the individual, well-placed vertebrae meshes are deformed in a free-form manner to adapt to their respective target bones.

As a result, the accuracy of the segmentation is improved by coupling individual objects according to anatomically reasonable, relative poses. However, like the previous approach this one depends on training data representing the full spectrum of different joint postures.

2.3 Explicit Joint Models for SSMs

A different idea was presented by [KLZH09] for the segmentation of the human hip. The proposed model is a compound SSM consisting of pelvis and thighbone. These bones are connected by an explicitly modeled hip joint. That is, joint variation is realized as parametrized rotation around the joint center.

In contrast to the approach of [HH95], the hip joint model separates variation of shape from variation of pose, which are indeed two different aspects. And unlike [KWLFO08], it integrates both aspects into one unified model. Another major difference to the previously mentioned methods is that pose variation is modeled explicitly rather than captured from training data. This allows for relative poses not present in the training data, but requires a joint implementation that reproduces anatomically reasonable joint motion.

The work of [KLZH09] can be seen as the origin of a joint effort by the Medical Planning group of the Zuse Institute Berlin. With [BKSLZH11], the author presented another explicit joint model for the human knee.

The work at hand is a continuation of this effort as it presents a general framework for models similar to, but more comprehensive than the models of the hip and knee joints. A preview on this thesis has been given in [BKRLZ12]. This approach enables models that consist of several objects connected by multiple joints. To this end, a comprehensive model of articulation is presented that adapts concepts from the field of multibody kinematics as they are described in [CD08] and [Sha05], for example.

2.4 Modeling Natural Joint Posture Variation

An integral aspect of this work are the joint models that realize joint posture variation. They must approximate the actual degrees of freedom of natural joints.

The model employed for the human hip is implemented according to the model presented in [KLZH09]. Regarding the human knee, [Fre01] provides a description of its complex mechanics. It serves as a basis for the simplified knee model that is used in the course of this work.

In order to build ASSMs that include joints apart from hip and knee, one must implement respective joint models. For this purpose, the reader is referred to [ETDH07] which describes methods to derive functional joint axes from analysis of skin marker data during natural motion.

3 Foundations

3.1 Representation of Shape

There are various ways to describe shapes in three-dimensional space. The methods discussed here describe shape in terms of their outer boundary (surface), discretely approximated by polygonal surface meshes. However, the concepts presented here may be applied to other representations as well.

The fundamental elements of the surface mesh representation are vertices (i.e. points in Euclidean space) and their connection by edges and faces to constitute a surface as shown in Figure 3.1.

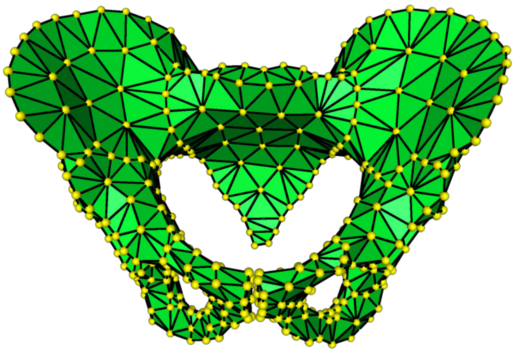


Figure 3.1: Elements of a surface mesh: Vertices (yellow), edges (black), triangular faces (green)

3.2 Representation of Spatial Pose

Rigid transformations are a versatile and effective means to describe and alter the spatial pose of geometric objects that are described by a set of points in Euclidean space. They comprise rotation and translation of a set of points relative to some frame of reference, e.g. a fixed world coordinate system.

The applications discussed in this work also require the variation of size of an object which may be realized by uniform, that is isotropic, scaling.

These aspects are implemented by application of transformation matrices to the vertices of an object. In order to enable translations, which are nonlinear in $\mathbb{R}^3$, points must be given in homogeneous coordinates $\in \mathbb{R}^4$. The transformation matrices are then elements of $\mathbb{R}^{4\times 4}$. In the course of this work, points may be described in either Euclidean or homogeneous coordinates, depending on the context. The representations can be converted according to the formulas provided in Appendix A.1.

More complex transformations can be composed by concatenation of multiple basic transformations. This is realized by the non-commutative matrix product denoted by the $\circ$ operator. Also, transformations can be inverted by inversion of the respective matrix. All transformations considered in this work are invertible as shown in Appendix A.2.

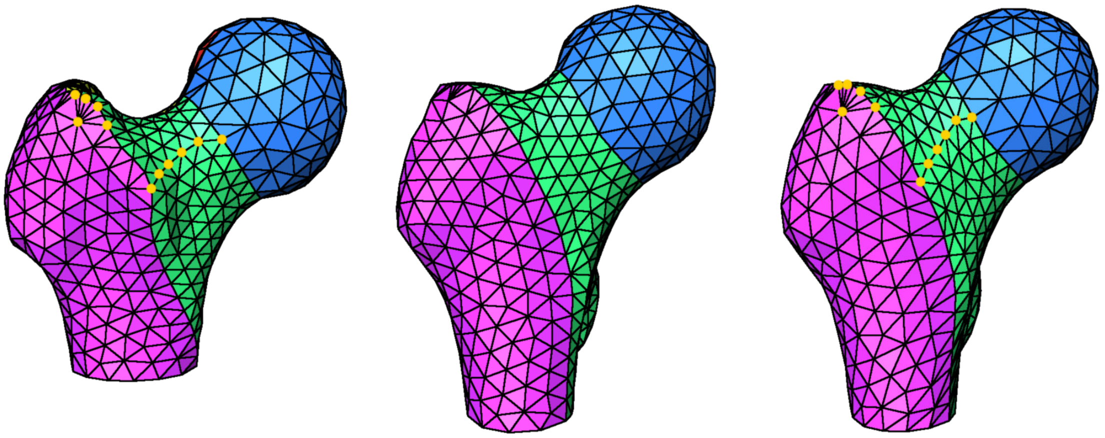
3.3 Prerequisites to Shape Analysis

SSMs are built by analysis of similar shaped instances of one object. To this end, it is important that these instances meet two prerequisites: Correspondence and alignment.

3.3.1 Correspondence

In a discrete representation, shape is described as a set of vertices and their connectivity. Since it is required to analyze multiple shape instances, correspondence between the vertex sets of different instances must be established. That is, vertices with the same index in different instances have to represent the same shape features. Furthermore, the connectivity between instances is required to remain constant.

Figures 3.2a and 3.2b show differently shaped parts of the thighbone with different meshes. Figure 3.2c depicts the second shape in terms of a mesh that corresponds to the first shape's mesh.



(a) Shape 1

(b) Shape 2

(c) Shape 2 in terms of a mesh that corresponds to that of shape 1

Figure 3.2: Correspondence using the example of the femoral head, the yellow points exemplify corresponding vertices

In general, the correspondence problem is hard to solve and still subject of current research. In this work, the method presented by [LSHD04] is used to establish correspondence. This approach consists of two stages. During the interactive stage, a reference object is decomposed into patches homeomorphic to discs. Then this patch layout is transferred to the target object. In the second and automatic stage, the target is remeshed patchwise with correspondence to the reference.

3.3.2 Alignment

The shape of an object is independent of its location and orientation. Moreover, an object's size is often not considered part of its shape. For a thorough definition and a discussion about size being a part of shape, see [DM98].

For a discrete representation of shape by vertices relative to some origin, these aspects must be removed to analyze different object instances. For this purpose, the instances are superimposed (aligned) with respect to a common frame of reference.

Given a reference instance represented by its vertices u_i, a corresponding instance with vertices v_i is aligned by minimization of the mean-square distance under a transformation T

$$\arg\min_{T} \sum_{i=1}^{N} \left| T \circ v_i - u_i \right|^2$$

with respect to a rigid transformation T, optionally including a uniform scale, i.e. size, component. This is a modification of the Procrustes method. For the details of implementation see [CTCG95; Goo91].

3.4 Statistical Shape Models

As already stated in Section 1.2 this work extends statistical shape models (SSMs) by the aspect of articulation. This section provides more details of those models to establish a foundation for the methods presented in Chapter 4.

An SSM represents the mean shape of an object and its variation in shape within a given training set by application of a point distribution model. The intention is to have a compact description of shape within a given shape space. This way, an SSM can estimate an entire, anatomically sound shape from partial or unclear target shape information.

3.4.1 Point Distribution Model

The point distribution model (PDM) incorporates information about a set of vertices, their average position and deviation regarding a set of training shapes.

To be able to analyze the training shapes, they must meet the criteria described in Section 3.3.

Given an aligned set of corresponding training shapes, each of it can be described as an element of a common vector space $\mathbb{R}^{3N}$ where N is the number of vertices approximating the shape. Principal component analysis (PCA) on the training set then yields a linear model

$$V(b) = \bar{V} + Pb$$

with the mean shape $\bar{V} \in \mathbb{R}^{3N}$, the matrix of eigenvectors of the covariance matrix $P \in \mathbb{R}^{3N \times K}$ and the shape weights $b \in \mathbb{R}^{K}$. Due to the nature of the applied method, the number of shape modes K equals the number of training shapes minus one.

The i-th vertex of that model in $\mathbb{R}^3$ is

$$v_i\left(b\right) = \left(V_{3i}\left(b\right), V_{3i+1}\left(b\right), V_{3i+2}\left(b\right)\right)^T \tag{3.1}$$

For thorough explanations on PCA and PDM, the reader is referred to [CTCG95; Shl05]. Furthermore, [DM98] is a comprehensive text book that describes these methods in the context of statistical shape analysis.

3.4.2 Varying Pose

A rigid transformation T may be applied to the vertices of the model in order to align it with any shape information (see Section 3.4.3). Optionally, T includes a uniform scale component. This adds 6 to 7 new degrees of freedom to the model and Equation (3.1) is extended to

$$\begin{aligned} v_i\left(b, T\right) &= T \circ v_i\left(b\right) \\ &= T \circ \left(V_{3i}\left(b\right), V_{3i+1}\left(b\right), V_{3i+2}\left(b\right), 1\right)^T \end{aligned}$$

3.4.3 Fitting the Model to a Target Anatomy

To fit an SSM with its vertices v_i to a target anatomy, the target has to be described as a corresponding set of vertices u_i. For arbitrary targets, these vertices can be found as points on the target surface that have a minimal distance to the points v_i.

Then the problem can be formulated as a weighted least-squares optimization

$$\underset{b,T}{\arg\min} \sum_{i=1}^{N} \left|v_i\left(b, T\right) - u_i\right|^2 w_i \tag{3.2}$$

with optional intermediate updates of u_i. The weights $w_i \in \mathbb{R}$ are used to favor target vertices with higher confidence over less certain targets.

As shown in [CTCG95; LSHD04], the optimization of Expression (3.2) can be divided into separate optimizations. First, the transformation is adjusted as described in Section 3.3.2, followed by the deformation of shape. To prohibit nonnatural shapes, the resulting shape weights are restricted to extremal values as derived from the training set. These steps are iterated until convergence to reach the best fit.

Note that an SSM's ability to approximate a particular target anatomy depends on its training data and especially the size of its training set.

3.5 Segmentation of Medical Image Data

Statistical shape models play a decisive role in the segmentation of volumetric images of intensities, like CT or MR images. This section gives a brief overview of an established segmentation pipeline that comprises following steps:

1. Detection of the object of interest in the image to initially position the SSM
2. Adaption of shape and pose of the SSM to the anatomy depicted by the image
3. Final adaption of the mesh by leaving the SSM shape space

The last step will not be described in detail here since it is beyond the scope of this work. For more information on this topic, see [SKHLZH08].

3.5.1 Pose Initialization

First of all, the object of interest must be located within the given image data. To this end, the Generalized Hough Transform (GHT), which is described in [Bal81], is used to roughly determine its pose.

Basically, this method searches the image volume exhaustively for a pose that may be a good initialization for the SSM. To this end, a rigid, coarsely grained mesh of the SSM's average shape is used to discretely search the low-dimensional parameter space of possible transformations while accumulating a best-fit measure. This measure may be the number of the mesh's surface normals that match the image gradient's directions. The configuration with the best fit is the assumed to be a good initialization of the SSM.

3.5.2 Determining Exact Pose and Shape

During the next stage, an SSM is used to approximate pose and shape of the anatomy inherently being described by given image data. The SSM is initialized according to the result of the GHT to avoid getting stuck in local minima during this optimization. Algorithm 3.1, which is taken from [SKHLZH08], deforms and repositions the SSM to fit the target anatomy in the image.

Algorithm 3.1 Iterative SSM-based segmentation

Input: Image I; initial model pose T_{GHT}; model $v_i(b) : \mathbb{R}^K \rightarrow \mathbb{R}^3$; stop criterion ϵ
Output: Shape weights b and pose T of the fitted model

```
1:  $T \leftarrow T_{GHT}$                                        ▷ Pose initialization after GHT
2:  $b \leftarrow 0$                                              ▷ Initially no deformation
3:  repeat
4:      $T' \leftarrow T$
5:      repeat
6:          $T'' \leftarrow T$
7:          $\forall i : (u_i, w_i) \leftarrow \textsc{ImageAnalysis}(I, v_i(b, T))$
8:          $T \leftarrow \arg\min_T \sum_i \left| v_i(b, T) - u_i \right|^2 w_i$          ▷ Pose update
9:      until $\sum_i \left| v_i(b, T) - v_i(b, T'') \right|^2 < \epsilon$          ▷ Until convergence
10:     $b' \leftarrow b$
11:     $\forall i : (u_i, w_i) \leftarrow \textsc{ImageAnalysis}(I, v_i(b, T))$
12:     $b \leftarrow \arg\min_b \sum_i \left| v_i(b, T) - u_i \right|^2 w_i$              ▷ Shape update
13: until $\sum_i \left| v_i(b, T) - v_i(b', T') \right|^2 < \epsilon$               ▷ Until convergence
```

The algorithm repeatedly analyzes the image data in the neighborhood of the currently positioned model to determine a new target u_i for each vertex v_i that better fits the image (lines 7 and 11). The displacement weights w_i are used to favor reliable targets over those located in regions that exhibit only unclear image features. According to Section 3.4.3, the adaption of the SSM w.r.t. pose (line 8) and shape (line 12) approximates a reasonable anatomy in such critical regions. The updates of pose and shape are aborted when the rate of change of the model's vertices fall below a user-defined criterion ϵ (lines 9 and 13).

A description and evaluation of applied medical segmentation methods as well as detailed information about the analysis of the image for displacements can be found in [SKHLZH08; LSHD04] and in the references therein.

4 A Framework for Articulated Statistical Shape Models

This chapter describes a generalized framework for articulated statistical shape models (ASSMs). To this end, a model of articulation is presented in the course of the next section. Then the articulation model and the point distribution model – the basis of SSMs – are combined to form ASSMs. The remainder of this chapter covers the application of those models to the problem of medical image segmentation. For that purpose, the segmentation methods of the previous chapter are adapted to handle the additional aspect of articulation.

4.1 Articulation Model

The articulation model describes individual poses of multiple objects in a compound in which the objects are connected by joints. The articulated connection of two objects imposes a restriction on the object poses: Since the pose of one object with respect to the other one shall reflect an anatomically reasonable joint posture, the individual poses must be coupled in terms of a joint's natural degrees of freedom.

To this end, the articulation model follows established methods of multibody kinematics developed in the field of robotics. Detailed descriptions of these methods can be found in [Sha05; CD08]. To realize the above requirements in terms of these methods, the articulation model consists of

- a description of the object-joint-connectivity of the compound that is presented in the next section,
- an algorithm that enables the derivation of kinematic chains from this connectivity information and solves the forward kinematic problem of adjusting joint posture (see Section 4.1.3) and

- a method that formulates and solves the inverse kinematic problem, that is, the fit of object poses to corresponding target positions (see Section 4.1.4).

4.1.1 A Compound of Objects with Joints

The object-joint-connectivity can be described as a graph: The objects are its nodes and joints are represented as directed edges between adjacent objects.

The i-th object's pose is expressed as a rigid transformation T_i, however it may be desirable to allow a uniform scaling, as mentioned in Section 3.4.2.

To couple object transformations in an anatomically reasonable manner, a joint (i.e. edge) between two adjacent objects relates their transformations to each other. To this end, the k-th joint's posture is expressed as a parametrized, relative, rigid transformation $J_k(\alpha_k)$ with α_k being a vector that describes a joint's natural degrees of freedom.

A basic example is the connection of two objects, 1 and 2, with poses represented by transformations T_1 and T_2. The objects are connected by a joint with its relative transformation $J_1(\alpha_1)$. Figure 4.1 illustrates this example. Given the pose T_1 and a joint model J_1 with its parameters α_1, the other object's pose is $T_2 = T_1 \circ J_1(\alpha_1)$. $\circ$ is the non-commutative concatenation of transformations.

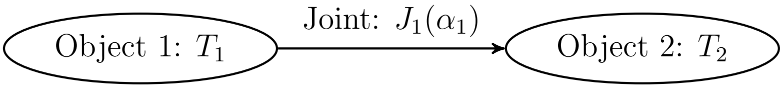


Figure 4.1: Two objects connected by a joint

Employing the graphical representation, it is possible to describe more complex compounds. Figure 4.2 shows an example where the objects 2, 3 and 4 are connected to the first object by the joints J_1, J_2 and J_3.

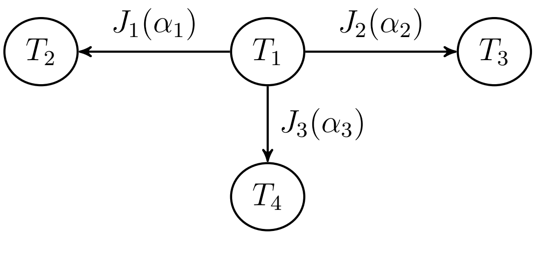


Figure 4.2: A more complex compound of connected objects

Limitation to Connectivity Trees

The pose T_2 of an object connected to another object 1 by joint $J_1(\alpha_1)$ is uniquely determined by T_1, J_1 and α_1. To prevent ambiguity, there must not exist any other dependency between the poses T_1 and T_2. Hence, the connection graph must not contain any cycles – it is indeed a tree.

Formal Description of Connectivity Trees

For a compound of M objects, the nodes of the tree are object indices $i \in \{1, \dots, M\}$ that also map to the respective transformations T_i.

The directed edges $(n, m) \in \{1, \dots, M\} \times \{1, \dots, M\}$ map to relative, rigid joint transformations J_k, which are explained in Section 4.1.6. The head of an edge points to object m whose pose is dependent on the respective joint's transformation and object n's pose.

A *Fixed* Object as Root of the Tree

So far, the transformation of an object is determined by its articulated connection to another object. That is, there has to be one particular object all the other objects refer to. This object is said to be *fixed* and it is not transformed relative to another object of the compound.

Furthermore, the *fixed* object may be regarded as the root of the connectivity tree. To make all other objects directly or indirectly dependent on the root, one must possibly flip some edges. These object dependencies can be inverted by inversion of the joint transformations. That is, an edge $(n, m) \to J_k$ can be converted to $(m, n) \to J_k^{-1}$, as shown in Figure 4.3. This allows for an arbitrary selection of the *fixed* object of the compound.

T_1
$J_1(\alpha_1)$
T_2
T_1
$J_1^{-1}(\alpha_1)$
T_2

Figure 4.3: Joint inversion, left and right joint specifications express the same relation between T_1 and T_2

4.1.2 Varying Pose of the Object Compound

The alignment of the whole object compound with any external shape information, e.g. a reference anatomy, is necessary for many applications. To this end, a pseudo-object representing the compound environment is attached to the *fixed* root object. Its transformation is the identity I and the edge to the root is associated with a *global transformation* $T_g(\alpha_g)$, as depicted in Figure 4.4.

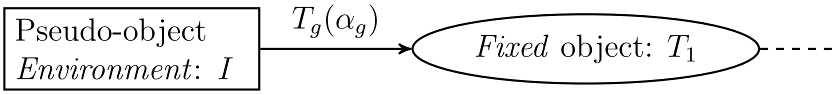


Figure 4.4: Introduction of the environment pseudo-object

T_g is a rigid transformation, optionally including a uniform scaling. Hence, it adds three translational components, three rotational components and one optional scaling component to the model's degrees of freedom.

4.1.3 Determining Individual Object Transformations

The previous sections have introduced the object-joint-connectivity of an articulated compound of objects. Now this information is used to address the forward kinematic problem, that is, kinematic chains are derived to determine the individual object transformations with respect to the compound's *global transformation* $T_g(\alpha_g)$ and the joint postures $J_k(\alpha_k)$ or $J_k^{-1}(\alpha_k)$, respectively.

To this end, a *fixed* object must have been declared and the *global transformation*, as described in Section 4.1.2, must have been attached to it. By application of Algorithm 4.1 the graph is traversed to concatenate the individual transformations. Appendix B.1 illustrates this algorithm with an example.

The algorithm calls the procedure TraverseGraph(i, p, T) recursively for every object of the graph, with the current object i, the previously visited object p and the current transformation T.

It starts at the pseudo-object *Environment* with the identity transformation and no predecessor in Line 12. Line 2 stores the current transformation T associated with the current object i. The current object's edges to not yet visited neighbors are

Algorithm 4.1 Determining object transformations from the connectivity graph

Input: Edges (directed from f to t) and associated transformations T_e as (f, t, T_e)-triples $\in E$; for each T_e also: T_e^{-1}

Output: Object transformations T_i

```
 1: procedure TraverseGraph($i, p, T$)
 2:     $T_i \leftarrow T$                                    ▷ Store result
 3:     for all $(i, n, T_e) \in E, n \neq p$ do              ▷ Traverse forward edges
 4:         $T' \leftarrow T \circ T_e$
 5:         TraverseGraph($n, i, T'$)
 6:     end for
 7:     for all $(n, i, T_e) \in E, n \neq p$ do              ▷ Traverse backward edges
 8:         $T' \leftarrow T \circ T_e^{-1}$
 9:         TraverseGraph($n, i, T'$)
10:     end for
11: end procedure

12: TraverseGraph($Environment, \emptyset, I$)
```

traversed in Lines 3 and 7 and the according, relative transformation T_e, respectively its inverse, is appended to the current transformation in Lines 4 and 8.

Hence, resulting object transformation T_i is a kinematic chain comprising the *global transformation* T_g and several, sometimes inverted, joint transformations J_k:

$$T_i(\alpha) = T_g(\alpha_g) \circ J_x^{t(x)}(\alpha_x) \circ J_y^{t(y)}(\alpha_y) \circ \cdots \circ J_z^{t(z)}(\alpha_z) \tag{4.1}$$

with

$$t(k) = \begin{cases} 1 & \text{if edge } k \text{ was traversed forwards} \\ -1 & \text{if edge } k \text{ was traversed backwards} \end{cases}$$

The transformation parameter $\alpha \in \mathbb{R}^q$ contains all q degrees of freedom of the respective articulation model. It is a concatenation of the *global transformation* parameter α_g and all joint parameters $\{\alpha_k\}$. The mapping $\alpha \leftrightarrow (\alpha_g, \{\alpha_k\})$ is a simple practical concern.

4.1.4 Fitting Object Poses to a Target Anatomy

To fit object poses of an articulated compound to corresponding objects of a target anatomy, the inverse kinematic problem must be solved. After *fixing* one object of the compound and determining the object transformations T_i from joint transformations J_i and compound pose transformation T_g, the fit to target vertices u_{ij} is a weighted least-squares minimization:

$$\arg\min_{\alpha} d(\alpha) \tag{4.2}$$

with

$$d(\alpha) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| T_i(\alpha) \circ v_{ij} - u_{ij} \right|^2 w_{ij} \tag{4.3}$$

Here, two indices are used to address the individual vertices. Index i enumerates the M compound objects while j enumerates the object's vertices: v_{ij} is the j-th vertex of object i, u_{ij} is its corresponding target position and w_{ij} weights its displacement.

Solving the Optimization Problem

In general, the object transformations $T_i(\alpha)$ are nonlinear w.r.t. α. That is, Expression (4.2) can not be calculated analytically, but must be approximated by nonlinear optimization. This process benefits from explicit information about the gradient of the objective function given in Equation (4.3):

$$\nabla_\alpha d(\alpha) = \left(\frac{\partial}{\partial \alpha_1}, \frac{\partial}{\partial \alpha_2}, \ldots, \frac{\partial}{\partial \alpha_q} \right) d(\alpha) \tag{4.4}$$

with

$$\begin{aligned}
\frac{\partial d(\alpha)}{\partial \alpha_s} &= \frac{\partial}{\partial \alpha_s} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| T_i(\alpha) v_{ij} - u_{ij} \right|^2 w_{ij} \\
&= \sum_{i=1}^{M} \sum_{j=1}^{N_i} \frac{\partial}{\partial \alpha_s} \left(T_i(\alpha) v_{ij} - u_{ij} \right)^T \left(T_i(\alpha) v_{ij} - u_{ij} \right) w_{ij}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{M}\sum_{j=1}^{N_i}\left(\frac{\partial}{\partial\alpha_s}v_{ij}^T T_i^T T_i v_{ij} - 2\frac{\partial}{\partial\alpha_s}u_{ij}^T T_i v_{ij} + \underbrace{\frac{\partial}{\partial\alpha_s}u_{ij}^T u_{ij}}_{=0}\right)w_{ij} \\
&= \sum_{i=1}^{M}\sum_{j=1}^{N_i}\left(v_{ij}^T\left(\frac{\partial}{\partial\alpha_s}T_i^T T_i\right)v_{ij} - 2u_{ij}^T\frac{\partial T_i}{\partial\alpha_s}v_{ij}\right)w_{ij} \\
&\overset{\text{Eqn.}}{\underset{\text{(A.1)}}{=}} \sum_{i=1}^{M}\sum_{j=1}^{N_i}\left(2v_{ij}^T T_i^T\frac{\partial T_i}{\partial\alpha_s}v_{ij} - 2u_{ij}^T\frac{\partial T_i}{\partial\alpha_s}v_{ij}\right)w_{ij} \\
&= 2\sum_{i=1}^{M}\sum_{j=1}^{N_i}\left(v_{ij}^T T_i^T\frac{\partial T_i}{\partial\alpha_s}v_{ij} - u_{ij}^T\frac{\partial T_i}{\partial\alpha_s}v_{ij}\right)w_{ij} \\
&= 2\sum_{i=1}^{M}\sum_{j=1}^{N_i}\left(v_{ij}^T T_i^T - u_{ij}^T\right)\frac{\partial T_i}{\partial\alpha_s}v_{ij}\cdot w_{ij} \qquad (4.5)
\end{aligned}
$$

where $\partial T_i/\partial\alpha_s$ is the elementwise partial derivative of T_i with respect to α_s. The calculation of this matrix of derivatives is explained in Section 4.1.5.

Efficient Computation of the Objective Function and its Gradient

The evaluation of the objective function and its gradient, as described in Equations (4.3) and (4.5), can be computationally expensive. However, repeated summation over an object's vertices can be avoided when the vertices, e.g. the shape, remain constant. By storing intermediate values for substitution, the number of iterations of this loop can be reduced to one. This approach is described further in Appendices A.5 and A.6.

4.1.5 Determining Partial Derivatives of Object Transformations

As shown in Section 4.1.3, the object transformations are concatenations of one parametrized global and several parametrized joint transformations:

$$T_i(\alpha) = T_g(\alpha_g) \circ J_x^{d(x)}(\alpha_x) \circ J_y^{d(y)}(\alpha_y) \circ \cdots \circ J_z^{d(z)}(\alpha_z)$$

with $d(k)$, equal to 1 or -1, either indicating an original or inverted matrix J_k.

According to [Min00], the chain rule for matrices can be applied to write $\partial T_i / \partial \alpha_s$ as:

$$\begin{aligned}
\frac{\partial T_i(\alpha)}{\partial \alpha_s} &= \frac{\partial}{\partial \alpha_s} T_g(\alpha_g) \circ J_x^{d(x)}(\alpha_x) \circ J_y^{d(y)}(\alpha_y) \circ \cdots \circ J_z^{d(z)}(\alpha_z) \\
&= \frac{\partial T_g(\alpha_g)}{\partial \alpha_s} \circ J_x^{d(x)}(\alpha_x) \circ J_y^{d(y)}(\alpha_y) \circ \cdots \circ J_z^{d(z)}(\alpha_z) \\
&\quad + T_g(\alpha_g) \circ \frac{\partial J_x^{d(x)}(\alpha_x)}{\partial \alpha_s} \circ J_y^{d(y)}(\alpha_y) \circ \cdots \circ J_z^{d(z)}(\alpha_z) \\
&\quad + T_g(\alpha_g) \circ J_x^{d(x)}(\alpha_x) \circ \frac{\partial J_y^{d(y)}(\alpha_y)}{\partial \alpha_s} \circ \cdots \circ J_z^{d(z)}(\alpha_z) \\
&\quad + \ldots\ldots \\
&\quad + T_g(\alpha_g) \circ J_x^{d(x)}(\alpha_x) \circ J_y^{d(y)}(\alpha_y) \circ \cdots \circ \frac{\partial J_z^{d(z)}(\alpha_z)}{\partial \alpha_s}
\end{aligned}$$

This equation can be simplified since at most one of the chained transformations does actually depend on α_s. That is, all summands that include a derived transformation not dependent on α_s equal the zero matrix and only the summand with the derived transformation that depends on α_s remains.

Considering, for example, the partial derivative w.r.t. a parameter of J_x, here $\alpha_{x,s}$, the above equation can be reduced to

$$\frac{\partial T_i(\alpha)}{\partial \alpha_{x,s}} = T_g(\alpha_g) \circ \frac{\partial J_x^{d(x)}(\alpha_x)}{\partial \alpha_{x,s}} \circ J_y^{d(y)}(\alpha_y) \circ \cdots \circ J_z^{d(z)}(\alpha_z)$$

Algorithm 4.2 implements the concatenation of global and optionally inverted joint transformations and their derivatives, respectively, to determine the derivative of an object transformation.

It works similar to Algorithm 4.1 as it traverses the connectivity graph to concatenate transformations. But when it processes the node that maps to the α_s-dependent transformation the respective derivative or its inverse is appended rather than the transformation itself (see Lines 8 and 16). Before the resulting derivative for an object is stored the current transformation chain is tested whether it contains a derived term (Line 2). If this is not the case the whole expression does not depend on α_s and the derivative matrix is the zero matrix (Line 5).

Algorithm 4.2 Determining partial derivatives of object transformations from the connectivity graph

Input: Edges (directed from f to t) and associated transformations T_e as (f, t, T_e)-triples $\in E$; for each T_e also: T_e^{-1}, $\partial T_e / \partial \alpha_s$, $\partial T_e^{-1} / \partial \alpha_s$

Input: Variable to differentiate with respect to, α_s

Output: Derivatives of object transformations $\partial T_i / \partial \alpha_s$

```
procedure TraverseGraph($\alpha_s, i, p, T$)
    if $T$ contains derivative then                      ▷ Store result $\partial T_i / \partial \alpha_s$
        $\partial T_i / \partial \alpha_s \leftarrow T$
    else
        $\partial T_i / \partial \alpha_s \leftarrow \mathbf{0}$                    ▷ $T_i$ not dependent on $\alpha_s$
    end if

    for all $(i, n, T_e) \in E, n \neq p$ do                 ▷ Traverse forward edges
        if $T_e$ dependent on $\alpha_s$ then
            $T' \leftarrow T \circ \partial T_e / \partial \alpha_s$
        else
            $T' \leftarrow T \circ T_e$
        end if
        TraverseGraph($n, i, T'$)
    end for

    for all $(n, i, T_e) \in E, n \neq p$ do                 ▷ Traverse backward edges
        if $T_e$ dependent on $\alpha_s$ then
            $T' \leftarrow T \circ \partial T_e^{-1} / \partial \alpha_s$
        else
            $T' \leftarrow T \circ T_e^{-1}$
        end if
        TraverseGraph($n, i, T'$)
    end for
end procedure

TraverseGraph($\alpha_s, Environment, \emptyset, I$)
```

4.1.6 General Specification of Joint Models

A joint model represents the natural variation of posture of a particular joint type. Therefore, it must approximate a real joint's natural degrees of freedom and must simulate natural anatomical axes and landmarks guiding displacement and rotation. Those can be derived from the geometry of the connected objects by querying vertex coordinates of the ASSM.

In matters of the proposed model of articulation a model of the k-th joint maps n degrees of freedom onto a rigid, relative transformation expressed by a transformation matrix:

$$J_k : \mathbb{R}^n \to \mathbb{R}^{4\times 4}$$

As required by the method presented in the previous section, a joint model must also provide the transformation's elementwise partial derivative w.r.t. its degrees of freedom α_s

$$\frac{\partial J_k}{\partial \alpha_s} : \mathbb{R}^n \to \mathbb{R}^{4\times 4}$$

Given the joint transformation and its derivatives, the inversion of a joint can be accomplished by the inversion of its transformation matrix. See Section 3.2 for a discussion on the invertibility of the transformations employed in this work. According to [Min00], the partial derivatives of an inverted joint's transformation can be calculated by

$$\frac{\partial J_k^{-1}}{\partial \alpha_s} = -J_k^{-1} \circ \frac{\partial J_k}{\partial \alpha_s} \circ J_k^{-1}$$

Section 6.1 presents concrete implementations of hip and knee joint models. To create such models, knowledge about a joint's anatomy and particularly its degrees of freedom is an important prerequisite. To this end, a natural joint's functional axes may be determined by analysis of motion data – e.g. skin marker positions – as described in [ETDH07].

4.2 Prerequisites to Shape Analysis of Articulated Objects

As described in Section 3.3, there are some prerequisites that have to be met in order to build an statistical model from a set of given training shapes.

4.2.1 Correspondence

For articulated compounds, this prerequisite includes correspondence of objects and joints of the respective compounds and also corresponding surface meshes that describe the shape of objects. The former is achieved by consistent numbering of objects and joints. Then the object meshes can be processed individually as proposed in Section 3.3.1 to establish correspondence. Though, some vertices are used to define joint axes – these landmarks have to be treated carefully. It is advised to fix them during the first, interactive stage of the proposed method when the exact placement is manually supervised. For details, refer to [LSHD04].

4.2.2 Alignment

As in the preparation of training shapes for conventional SSMs, the individual compounds of a training set have to be aligned. That is, differences in position, orientation and size between the compounds have to be removed. In contrast to the approach discussed in Section 3.3.2, these aspects cannot be altered independently for each object of a compound since it would distort anatomically correct poses of objects relative to each other.

However, the articulation model proposed in Section 4.1 can be applied to the training compounds. The alignment can then be performed under variation of position, orientation and size of the whole compound and under anatomically reasonable variation of posture of respective joints. For this purpose, the fit described in Section 4.1.4 is applied to the individual compounds of the training set with respect to a reference pose. This reference may be chosen from the training set.

Furthermore, this approach may be iterated to avoid any bias towards the chosen reference. To this end, all training shapes may be aligned to some training compound. The average of all shapes may then be taken as reference for the next iteration of alignment. This step is then repeated until the changes between subsequent iterations fall below a user-defined threshold.

4.3 Point Distribution Model and Articulation Model for ASSMs

In order to build ASSMs, the articulation model (see Section 4.1.1) and the point distribution model (see Section 3.4.1) need to be combined. The following sections describe the adaption of the latter for the application to articulated compounds, the integration of variation of shape and pose and a method that fits an ASSM to a target anatomy w.r.t. both aspects.

4.3.1 Point Distribution Model for Articulated Compounds

Shape variation can be described by a point distribution model, as described in Section 3.4.1, which yields a linear model

$$V(b) = \bar{V} + Pb$$

To capture correlating shape variation of adjacent objects, principal component analysis is performed on the aligned set of training compounds. That is, P, $\bar{V}$ and $V(b)$ contain information about the vertices of all objects the compound consists of.

Extending Equation (3.1) to distinguish individual objects, the position of the j-th vertex of the i-th object is

$$v_{ij}(b) = (V_{\mathrm{idx}(i)+3j}(b), V_{\mathrm{idx}(i)+3j+1}(b), V_{\mathrm{idx}(i)+3j+2}(b))^T \tag{4.6}$$

with $\mathrm{idx}(i)$ being the index of the first component of the first vertex of the i-th object in the coordinate vector V.

4.3.2 Integration into an ASSM

The shape variation expressed by Equation (4.6) and the variation of object poses according to Equation (4.1) can be combined by the application of T_i to the vertices of object i.

However, variation of shape may change anatomical axes and landmarks which are used to define joint transformations. Thus, the former model of joint transformations $J_k(\alpha_k)$ must be altered to $J_k(\alpha_k, b)$. In turn, the formulation of the object transformations given in Equation (4.1) must be changed to

$$T_i(\alpha, b) = T_g(\alpha_g) \circ J_x^{t(x)}(\alpha_x, b) \circ J_y^{t(y)}(\alpha_y, b) \circ \cdots \circ J_z^{t(z)}(\alpha_z, b)$$

Then the j-th vertex of the i-th object is

$$v_{ij}(b, \alpha) = T_i(\alpha, b) \circ v_{ij}(b)$$

4.3.3 Fitting the Model to a Target Anatomy

The fit of an ASSM to a given object compound represented by a set of corresponding target vertices u_{ij} can be written as weighted least-squares optimization

$$\begin{aligned} &\arg\min_{b,\alpha} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| v_{ij}\left(b, \alpha\right) - u_{ij} \right|^2 w_{ij} = \\ &\arg\min_{b,\alpha} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| T_i(\alpha, b) \circ v_{ij}(b) - u_{ij} \right|^2 w_{ij} \end{aligned}$$

In analogy to the separation of concerns for conventional SSMs described in Section 3.4.3, this optimization is approximated by separation into two distinct steps. The fit of object poses is performed under invariant shape, that is constant b:

$$\arg\min_{\alpha} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| T_i(\alpha) \circ v_{ij} - u_{ij} \right|^2 w_{ij} \tag{4.7}$$

This subproblem is solved by the method presented in Section 4.1.4.

To simplify the shape fit the following assumption is made: Since joint axes and landmarks vary only slightly under variation of shape, the object poses T_i are assumed to be independent of b and thus constant. Then the adaption of shape can be written as

$$\arg\min_b \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left|T_i \circ v_{ij}(b) - u_{ij}\right|^2 w_{ij}$$

To solve this subproblem the vertices u_{ij} are transformed by T_i^{-1}, that is, into the coordinate space that is independent of object transformations. Then the optimization

$$\arg\min_b \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left|v_{ij}(b) - u'_{ij}\right|^2 w_{ij} \quad \text{with} \quad u'_{ij} = T_i^{-1} \circ u_{ij} \tag{4.8}$$

can be performed similar to the one for conventional SSMs (see Section 3.4.3).

The repeated, alternating minimization of the Expressions (4.7) and (4.8) is assumed to fit the compound to the target w.r.t. pose and shape and compensates for deviations introduced by the assumption of invariant joint axes and landmarks under shape variation. See Section 6.3.1 for an evaluation of this approach.

4.4 Application of ASSMs to Medical Image Segmentation

The SSM-based segmentation approach discussed in Section 3.5 must be altered slightly to be compatible with an articulated model. The following sections describe the modification of the original method.

4.4.1 Pose Initialization

The approximate poses of individual objects of a compound model can be determined independently by the GHT as presented in Section 3.5.1. However, there are multiple objects to be considered and knowledge about their relative poses to each other should be exploited. The following approach considers these characteristics:

1. Detect the pose T_0 of the *most distinctly shaped object* in image I:
   $T_0 \leftarrow \mathrm{GHT}(0, I)$

2. An object m that is adjacent to an already detected object n is searched in a reduced volume of the image where its presence is assumed:
   $T_m \leftarrow \mathrm{GHT}(m, \textsc{Reduce}_m(T_n, I))$
3. Repeat step 2 until all object poses are determined.

The subsequent invocations of the GHT during step 2 benefit from the reduction of the volume to be searched. First, the correct pose can be detected more reliably since there are less possibilities of misdetection. Second, the smaller volume accelerates detection speed.

Note that the choice of the *most distinctly shaped object* in step 1 depends on the structure to be segmented and human intuition. For example, in the course of this work the pelvis was chosen over femur and tibia because of its distinct shape in comparison with the latter two.

After determination of initial object positions using coarsely meshed template shapes, another set of template shapes that corresponds to the ASSM is employed. These shapes are positioned according to the GHT. Then the ASSM is initialized by fitting it to the templates as described in Section 4.1.4. Note that the object poses also depend on the represented shape. Thus, it is advised to rather fit pose *and* shape to the templates as proposed in Section 4.3.3.

4.4.2 Determining Exact Pose and Shape

Algorithm 4.3 fits an ASSM to the compound anatomy depicted by a volumetric image. It is a modification of Algorithm 3.1 which fits conventional SSMs.

It basically differs from the algorithm for SSMs in the expression of pose by its degrees of freedom α rather than by a single transformation T and in the initialization by fit of pose and especially shape in line 1 as described in the previous section.

The update of pose and shape in lines 7 and 11, respectively, are performed according to Section 4.3.3.

The analysis of the image in the neighborhood of the actual model instance (lines 6 and 10) is the same as for single SSMs. However, the particular characteristics of individual objects can be taken into account as the analysis for one type of object may differ from those for other types.

Algorithm 4.3 Iterative ASSM-based segmentation

```
Input: Image $I$; initially positioned templates with vertices $t_{ij}$; model $v_{ij}(b, \alpha) : \mathbb{R}^{K+q} \rightarrow \mathbb{R}^3$; stop criterion $\epsilon$
Output: Shape weights $b$ and pose vector $\alpha$ of the fitted model

(b, α) ← $\arg\min_{b,\alpha} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| v_{ij}(b, \alpha) - t_{ij} \right|^2$    ▷ Pose and shape initialization according to Section 4.4.1
repeat
    $\alpha' \leftarrow \alpha$
    repeat
        $\alpha'' \leftarrow \alpha$
        $\forall i, j : (u_{ij}, w_{ij}) \leftarrow$ ImageAnalysis$(I, v_{ij}(b, \alpha))$
        $\alpha \leftarrow \arg\min_{\alpha} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| v_{ij}(b, \alpha) - u_{ij} \right|^2 w_{ij}$    ▷ Pose update
    until $\sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| v_{ij}(b, \alpha) - v_{ij}(b, \alpha'') \right|^2 < \epsilon$    ▷ Until convergence
    $b' \leftarrow b$
    $\forall i, j : (u_{ij}, w_{ij}) \leftarrow$ ImageAnalysis$(I, v_{ij}(b, \alpha))$
    $b \leftarrow \arg\min_{b} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| v_{ij}(b, \alpha) - u_{ij} \right|^2 w_{ij}$    ▷ Shape update
until $\sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| v_{ij}(b, \alpha) - v_{ij}(b', \alpha') \right|^2 < \epsilon$    ▷ Until convergence
```

Section 6.3.2 evaluates the presented algorithm with the help of an example.

5 Implementation

5.1 Articulated Statistical Shape Models in Amira

The method presented in this work is realized in Amira, which is an extendable software system for scientific visualization, data analysis and presentation of 3D and 4D data. See [Ami] for more details.

The following section briefly introduces an existing implementation of SSMs as they are outlined in Section 3.4. Then the adaption towards ASSMs is explained.

5.1.1 Statistical Shape Models

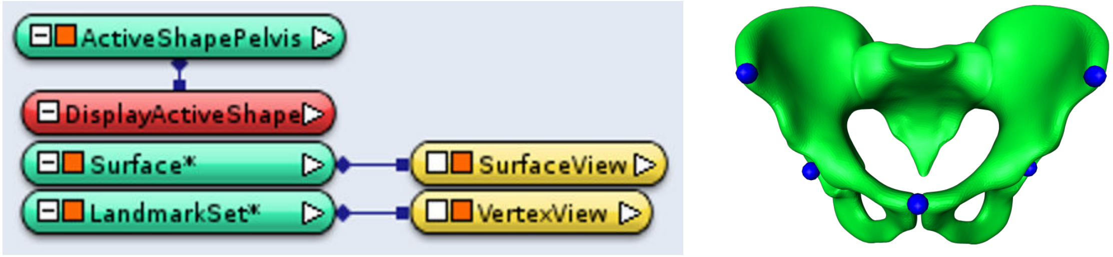
In Amira, the basic element of an SSM is the `ActiveShape` class. It comprises of

- the mean shape vector $\bar{V}$ and the matrix of eigenvectors of the covariance matrix P representing the point distribution model described in Section 3.4.1, as well as
- the shape vector b and a transformation T representing the model's state regarding shape and spatial pose (see Sections 3.4.1 and 3.4.2).

In addition, information about the vertices' connectivity are stored in a way that enables representation of multiple geometric objects of varying type, but with corresponding shape and pose.

The module `DisplayActiveShape` materializes an `ActiveShape` into a model instance of particular shape and pose. Therefore it generates geometries like surfaces, line- and landmark sets as intermediate data that can be processed and visualized.

Figure 5.1a depicts an example of such a network: The `DisplayActiveShape` generates a surface and a set of landmark points from the "ActiveShapePelvis" SSM as an instance of the model. These geometries are visualized by the yellow display modules. Figure 5.1b shows the visual output of Amira.



(a) Amira network that materializes and displays a particular state of an SSM

(b) SSM as displayed in Amira

Figure 5.1: An SSM in Amira

5.1.2 Articulated Statistical Shape Models

The implementation of ASSMs shall be able to seamlessly substitute SSMs. To this end, an ASSM must resemble the modular structure of conventional `ActiveShapes` and provide the same interface to modules that perform model adjustment.

To this end, it is possible to derive a new type for ASSMs from `ActiveShape` polymorphically. However, the introduction of a new type to Amira would involve the adaption of plenty existing tools. As a compromise a different approach is chosen: The current `ActiveShape` class is extended by an optional member that handles articulation – the `JointManager` object:

```
class ActiveShape {
    JointManager *jointManager;
    [...]
};
```

If the pointer `jointManager` equals `0` the `ActiveShape` is a conventional SSM, otherwise it represents an ASSM. Thus, SSMs and ASSMs exhibit a common interface that realizes basic functionality. Different behavior, e.g. regarding the adaption of object poses, is realized by delegation to methods of the `JointManager` class for ASSMs. Explicit access to functions that relate to articulation is offered by the getter method `ActiveShape::getJointManager()`.

Implementation of the Articulation Model

The `JointManager` class is the central element of ASSMs realized by `ActiveShape` objects. It implements the articulation model presented in Section 4.1. For this purpose, the `JointManager` class consists of

- a description of relations between objects and joints (see Section 4.1.1),
- a list of `Joint` objects that implement particular joint models (see Section 4.1.6),
- a member representing the global transformation (see Section 4.1.2),
- methods to determine object transformations (see Section 4.1.3),
- methods to fit object poses to a target anatomy (see Section 4.1.4) and
- a member vector that represents the value of the joint and global transformation parameter α.

The `DisplayActiveShape` module is modified to apply individual object transformations to the geometry objects it generates if a `JointManager` exists for a given model.

Generation of ActiveShape ASSMs

In Amira, SSMs are created by a particular module that processes a set of aligned, corresponding training shapes, computes the PCA and finally generates the model. Since this module is already able to handle non-articulated compounds of objects, it is used to create intermediate SSMs. In the course of this work, another module was written to setup the `JointManager` object and inject it into an SSM to turn it into an ASSM.

To generate an ASSM from a given set of unaligned training shapes that already correspond w.r.t. their objects and surface meshes, the following steps are processed:

1. A specific configuration of the articulation model is created, that is, the object-joint-connectivity graph is created and joint models are configured according to their placement in the graph. This finally yields a specific `JointManager` instance.
2. Each training compound is turned into an intermediate SSM and then into an intermediate ASSM by injection of the previously defined `JointManager`.

Since these models comprise only one shape, they do not realize any shape variation.

3. These intermediate ASSMs are used to align the individual training shapes to a reference pose chosen from the training set as described in Section 4.2.2 – by fitting the ASSM's pose to the reference as discussed in Section 4.1.4.
4. The now aligned training shapes are extracted from the individual ASSMs. They are now processed together by the SSM creation module. The resulting intermediate SSM now realizes shape variation according to the set of training compounds.
5. Finally this SSM is turned into an ASSM by injection of the `JointManager` created in step 1. The resulting model realizes the desired shape and joint posture variation.

The generation of the required, corresponding training meshes is carried out by Amira modules that implement the method discussed in Section 3.3.1.

5.1.3 Application of ActiveShapes to Image Segmentation

To apply an `ActiveShape` – representing an SSM or an ASSM – to the problem of segmentation, the Amira module `AdjustActiveShape` implements Algorithm 5.1 to fit the model to the anatomy depicted by an image. This algorithm unifies the Algorithms 3.1 and 4.3 suited to SSMs and ASSMs, respectively. It delegates the update of shape and pose to the `ActiveShape` which may either call the default implementation for SSMs or forward to the ASSM-implementation in the `JointManager` object.

Figure 5.2 depicts the network of modules in Amira that perform the `ActiveShape`-based image segmentation. The network consists of a three-dimensional `Image` to be segmented, an `ActiveShape` model to be adjusted including the `DisplayActiveShape` that materializes the geometries `Surface1`, `Surface2`, `...` and the `AdjustActiveShape` module that implements the adaption strategy.

Algorithm 5.1 Iterative `ActiveShape`-based segmentation

Input: Image I; initialized `ActiveShape` A with vertices V; stop criterion ϵ
Output: Model vertices V that approximate the depicted anatomy

```
repeat
    $V' \leftarrow V$
    repeat
        $V'' \leftarrow V$
        $(U, W) \leftarrow \textsc{ImageAnalysis}(I, V)$
        $V \leftarrow \text{A}.\textsc{UpdatePose}(U, W)$
    until $\sum_i \left|V_i - V''_i\right|^2 < \epsilon$        ▷ Until convergence
    $(U, W) \leftarrow \textsc{ImageAnalysis}(I, V)$
    $V \leftarrow \text{A}.\textsc{UpdateShape}(U, W)$
until $\sum_i \left|V_i - V'_i\right|^2 < \epsilon$        ▷ Until convergence
```

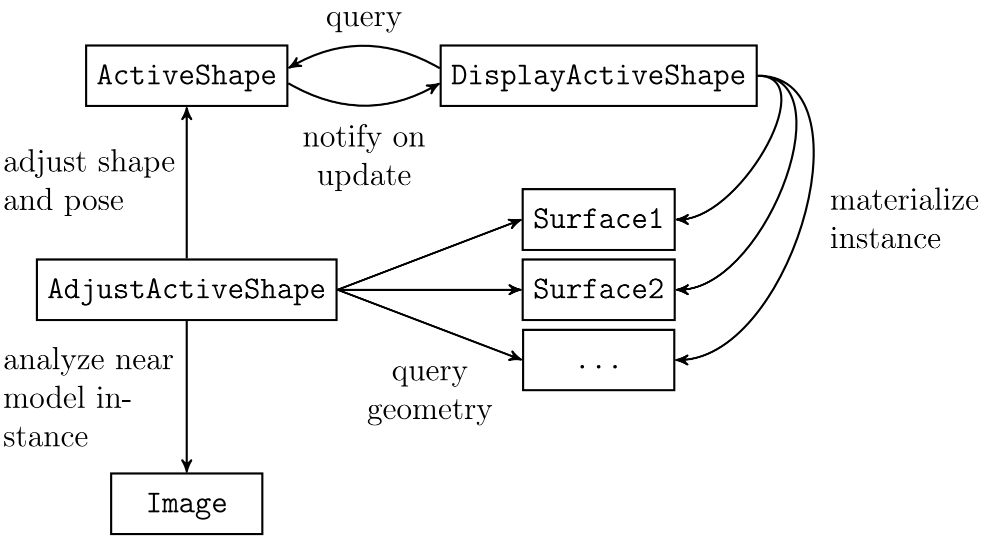


Figure 5.2: Segmentation in Amira: Fitting an `ActiveShape` to an image

5.2 Fitting Object Poses by Nonlinear Optimization

This section presents an implementation of the pose fitting problem described in Section 4.1.4. As already stated, this problem can be formulated as non-linear least-squares problem:

$$\arg\min_{\alpha} d(\alpha) \qquad \text{with} \qquad d(\alpha) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| T_i(\alpha) \circ v_{ij} - u_{ij} \right|^2 w_{ij}$$

Its solution requires a non-linear optimization strategy. To this end, this implementation employs `Ipopt`, a software library for large scale nonlinear optimization. A detailed description of its capabilities can be found in [Ipo] and [WB06].

The embedding of `Ipopt` is accomplished by deriving from `Ipopt::TNLP`, a base class for the implementation of objective functions. The resulting class must implement the evaluation of the objective function $d(\alpha)$ as well as its gradient $\nabla_\alpha d(\alpha)$. The value of $d(\alpha)$ can be computed directly, its gradient is determined by the proposed method in terms of Equations (4.4) and (4.5) in combination with Algorithm 4.2. Internally `Ipopt` also requires the Hessian matrix of the Lagrangian of the objective function. It can also be provided by the objective function object, but this implementation employs `Ipopt`'s quasi-Newton method to approximate the Hessian.

Furthermore, the objective function object must provide a starting point α_{init} for the iterative optimization performed by `Ipopt`. This point is given by the current state of the model.

`Ipopt` is fairly simple to use out of the box. However, its behavior can be tweaked by numerous options. For example, the optimization algorithm requires information about the objective function, its gradient and the Hessian of its Lagrangian. In this work, the option `hessian_approximation` is set to `limited-memory` to approximate the Hessian from the objective and its gradient.

Furthermore, the optimization abortion tolerance (`tol`) is set to 10^{-10} while the maximum number of iterations (`max_iter`) is limited to 500. The low tolerance requires `Ipopt` to iterate until a very close approximation of the optimum is found. Depending on the complexity of the specific ASSM in terms of number of objects and joints, such an accurate result may not be achieved quickly. However, it proved practical to abort the optimization after 500 iterations in such cases – the result is still accurate enough. So this configuration presents a reasonable compromise between optimization accuracy and computation time.

A list of options that are applied to `Ipopt` for the ASSM pose optimization is presented in Table 5.1.

Option	Value	Effect
`tol`	10^{-10}	Optimization stopping criterion ϵ
`max_iter`	500	Maximum number of iterations
`limited_memory_-max_history`	10000	Increase cache-size to reuse previously computed values
`mu_strategy`	adaptive	Performs automatic adjustment of the barrier term
`hessian_-approximation`	limited-memory	Enables quasi-Newton approximation of the Hessian

Table 5.1: `Ipopt` configuration for the optimization of object poses

Since the optimization is strongly influenced by the specific ASSM – and its complexity in particular – the present implementation allows to override these options for individual `ActiveShape` objects. For example, one may increase the abortion tolerance for ASSMs with plenty of joints that often reach the iteration limit during pose fitting.

6 Evaluation

This chapter shows the practical capabilities of the presented method. To this end, two articulated statistical shape models are presented: The toy example Boxman and a realistic model of the human lower limb. Both models are applied to the problem of image segmentation.

The Boxman ASSM consists of 12 articulated cuboids and serves as proof of concept for complex compounds that involve numerous joints.

The lower limb ASSM consists of pelvis, thighbone and shinbone which are connected by hip and knee. To evaluate the articulated approach with regards to medical image segmentation tasks, this model is applied to reconstruct anatomies from clinical CT images. The results are compared to reconstructions stemming from conventional single-object SSMs.

Since both ASSMs employ the same joint models, this chapter starts with a description of their implementations.

6.1 Joint Models

6.1.1 Model of Spheroidal Joints for the Human Hip

The model of the hip used here resembles the one presented in [KLZH09]. The human hip is an instance of a ball and socket joint, or spheroidal joint. This class of joints can be approximated as a rotation on an arbitrarily oriented axis through a fixed center. There are numerous ways to represent the three degrees of freedom of such a transformation: Euler angles, axis-angle pairs and unit quaternions are prominent examples.

In this work, Euler angles are favored over the others since they are straightforward to implement with regard to the resulting transformation and especially its optimization. But they suffer from the disadvantage of potential gimbal locks. Its consequences are discussed later in this section.

With Euler angles, a rotation around a fixed center is realized as three consecutive rotations around the axes of a coordinate system. There are several possibilities to compose the individual rotations. The present implementation rotates around the x-axis first, followed by rotations around the y- and then the z-axis. The axes are depicted in Figure 6.1. Note that the last two rotations also influence the axes of preceding rotations.

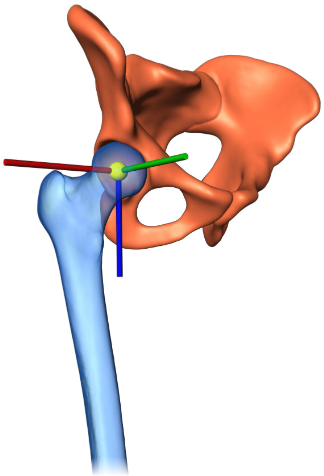


Figure 6.1: The spheroidal joint model that is used for the hip realizes three consecutive rotations around the depicted axes

Written as a relative, rigid transformation in terms of matrix concatenation the hip model is

$$J_{rot}(\beta) = \mathrm{transl}(\vec{c}) \circ \mathrm{rot}_z(\beta_z) \circ \mathrm{rot}_y(\beta_y) \circ \mathrm{rot}_x(\beta_x) \circ \mathrm{transl}(-\vec{c})$$

with rotation angles $\beta \in \mathbb{R}^3$. rot_i denotes a rotation around the i-th axis, $\vec{c} \in \mathbb{R}^3$ is the center of rotation and transl represents a translation. The effects of variation of β are depicted in Figure 6.2.

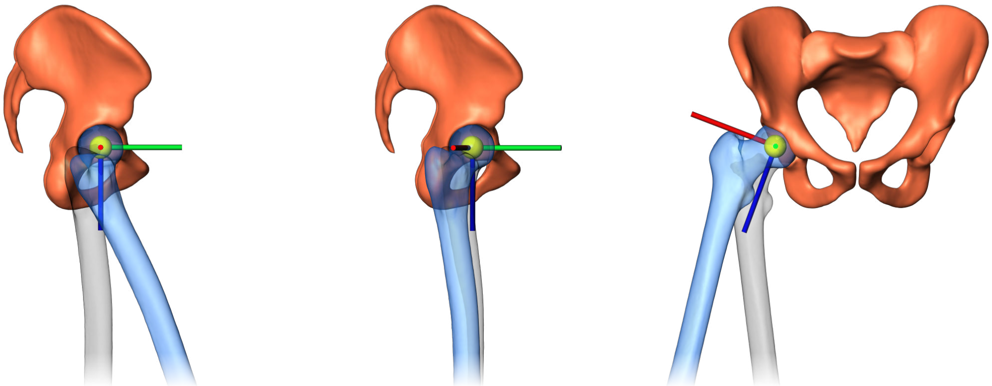
The partial derivatives w.r.t. β's components are

$$\frac{\partial J_{rot}(\beta)}{\partial \beta_x} = \mathrm{transl}(\vec{c}) \circ \mathrm{rot}_z(\beta_z) \circ \mathrm{rot}_y(\beta_y) \circ \frac{\partial\, \mathrm{rot}_x(\beta_x)}{\partial \beta_x} \circ \mathrm{transl}(-\vec{c})$$

$$\frac{\partial J_{rot}(\beta)}{\partial \beta_y} = \text{transl}(\vec{c}) \circ \text{rot}_z(\beta_z) \circ \frac{\partial\, \text{rot}_y(\beta_y)}{\partial \beta_y} \circ \text{rot}_x(\beta_x) \circ \text{transl}(-\vec{c})$$

$$\frac{\partial J_{rot}(\beta)}{\partial \beta_z} = \text{transl}(\vec{c}) \circ \frac{\partial\, \text{rot}_z(\beta_z)}{\partial \beta_z} \circ \text{rot}_y(\beta_y) \circ \text{rot}_x(\beta_x) \circ \text{transl}(-\vec{c})$$

The basic transformations and their respective derivatives are described in Appendix A.3.



(a) Rotation around x-axis (red)

(b) Rotation around z-axis (blue)

(c) Rotation around y-axis (green)

Figure 6.2: Variation of hip joint posture (default pose is outlined in gray)

As already mentioned, the representation of rotations by Euler angles introduces the possibility of gimbal locks. Such a lock emerges when the axes of first and last rotation are in a parallel configuration due to the second rotation. Such a situation may hamper the optimization of joint posture, but it can be avoided by restriction of permissible configurations. The composition of rotations applied here leads to a gimbal lock at $\beta_y = \pm\pi/2$. However, concerning the hip the rotation around the local y-axis represents abduction/adduction which can reasonably restricted to smaller angles (see Figure 6.2c).

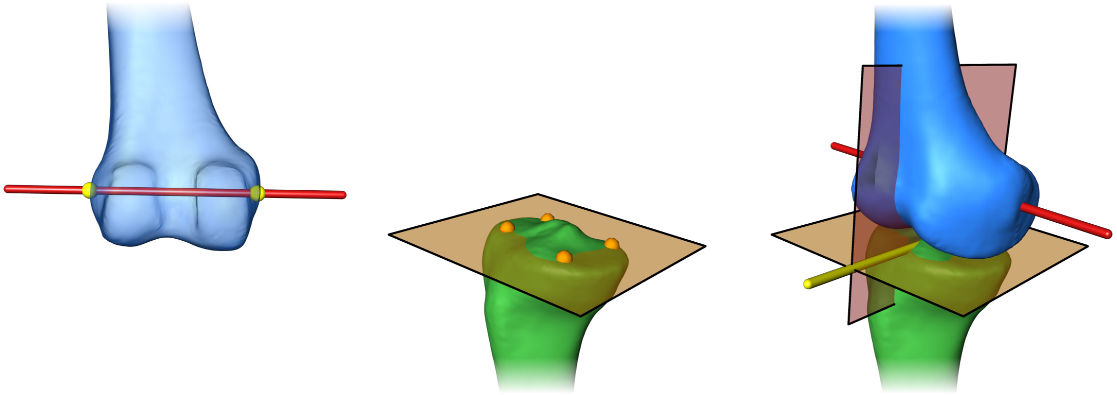
6.1.2 Model of the Human Knee

The human knee is a complex joint that actually consists of two parts: the patellofemoral component, which connects the kneecap (patella) to the thighbone (femur), and the tibiofemoral component, that articulates the femur with the shinbone (tibia). Since the available model of the lower limb does not include the patella, only the latter component of the knee joint is modeled. But even with the restriction to the tibiofemoral component the actual joint still comprises a complex mechanism of articulation as shown by [Fre01].

The simplified knee model that is employed here has already been published in [BKSLZH11]. It articulates the femur with the tibia under consideration of two degrees of freedom: The first realizes knee flexion while the second allows an anterior-posterior shift of the femur, that is forwards or backwards.

The knee flexion is simulated as rotation around the epicondylar axis. This axis connects two landmarks at the lateral and medial outmost positions on the surface of the lower femur as depicted in Figure 6.3a.

The anterior-posterior shift is guided along the line that intersects the epicondylar axis' normal plane with the tibial plateau. This plateau is defined as a plane fitted to a set of landmarks on top of the tibia as shown in Figure 6.3b. The construction of the actual translation axis is depicted in Figure 6.3c. Note that the origin points of respective planes are not relevant here – information about the planes' normal directions suffice to calculate the shift vector.



(a) Landmarks (yellow) that determine the epicondylar axis (red)

(b) Landmarks (orange) that determine the tibial plateau (yellow plane)

(c) Intersection of epicondylar axis' normal plane (red) and tibial plateau plane (yellow) constitute the anterior-posterior shift axis (yellow)

Figure 6.3: Knee joint landmarks and axes

The relative transformation of the knee model is

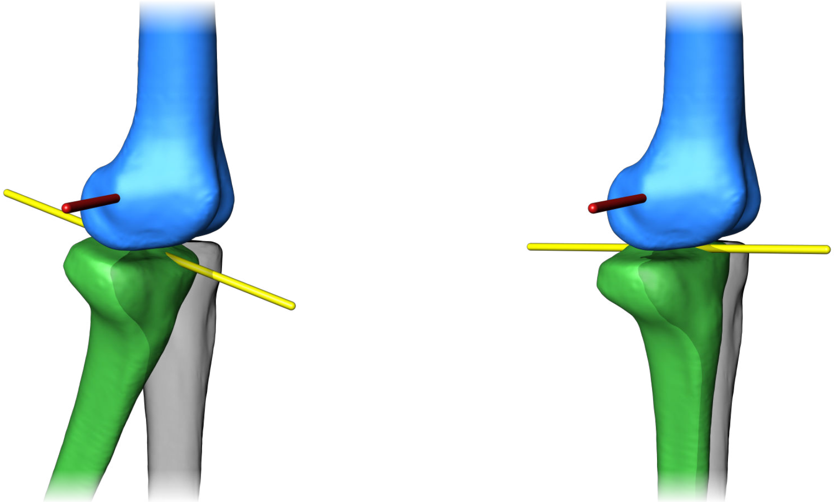
$$J_{knee}(\gamma, t) = \mathrm{transl}(\vec{c}_{epi}) \circ \mathrm{rot}_{\vec{d}_{epi}}(\gamma) \circ \mathrm{transl}(-\vec{c}_{epi}) \circ \mathrm{transl}_{\vec{d}_{epi} \times \vec{d}_{tib}}(t)$$

with rotation angle $\gamma \in \mathbb{R}$ and translation distance $t \in \mathbb{R}$. $\mathrm{rot}_{\vec{d}_{epi}}$ denotes a rotation around the epicondylar axis given by the directional unit vector $\vec{d}_{epi}$ and origin point $\vec{c}_{epi}$, $\vec{d}_{tib}$ is a unit normal of the tibial plateau. The partial derivatives of J_{knee} are

$$\frac{\partial J_{knee}(\gamma, t)}{\partial \gamma} = \mathrm{transl}(\vec{c}_{epi}) \circ \frac{\partial\, \mathrm{rot}_{\vec{d}_{epi}}(\gamma)}{\partial \gamma} \circ \mathrm{transl}(-\vec{c}_{epi}) \circ \mathrm{transl}_{\vec{d}_{epi} \times \vec{d}_{tib}}(t)$$

$$\frac{\partial J_{knee}(\gamma, t)}{\partial t} = \mathrm{transl}(\vec{c}_{epi}) \circ \mathrm{rot}_{\vec{d}_{epi}}(\gamma) \circ \mathrm{transl}(-\vec{c}_{epi}) \circ \frac{\partial\, \mathrm{transl}_{\vec{d}_{epi} \times \vec{d}_{tib}}(t)}{\partial t}$$

The basic transformations $\mathrm{rot}_{\vec{v}}$, $\mathrm{transl}_{\vec{v}}$, transl and their respective derivatives are described in Appendix A.3.



(a) Rotation by 20° around epicondylar axis (red axis)

(b) Translation by 1 cm along tibial plateau axis (yellow axis)

Figure 6.4: Variation of knee joint posture (default posture is outlined in gray)

6.2 Evaluation Measures

This section presents some measures to assess similarity of shapes. They are used in the remainder of this chapter to evaluate the presented method.

6.2.1 Measuring Distance of a Set of Surfaces to Target Surfaces

The accuracy of adaption of a surface to a target can be quantified in terms of residual distance to the target. To this end, the measures proposed in [LSHD04] are extended to consider multiple objects with respective targets.

Let

$$d(x, S) = \min_{x' \in S} |x - x'|$$

denote the distance of a point x to a surface S.

The mean distance of a set of surfaces $\mathcal{S}$ to corresponding targets $\mathcal{S}'$ can be written as

$$d_{mean}(\mathcal{S}, \mathcal{S}') = \frac{1}{\sum\limits_i (A(\mathcal{S}_i) + A(\mathcal{S}'_i))} \sum_i \left(\int\limits_{x \in \mathcal{S}_i} d(x, \mathcal{S}'_i) \,\mathrm{d}S + \int\limits_{x \in \mathcal{S}'_i} d(x, \mathcal{S}_i) \,\mathrm{d}S \right)$$

with i enumerating the individual surfaces, the area of a surface $A(\cdot)$ and $\mathrm{d}S$ being an infinitesimal area of the respective surface.

Analogously, the root-mean-square averaged distance is

$$d_{rms}(\mathcal{S}, \mathcal{S}') = \sqrt{\frac{1}{\sum\limits_i (A(\mathcal{S}_i) + A(\mathcal{S}'_i))} \sum_i \left(\int\limits_{x \in \mathcal{S}_i} d(x, \mathcal{S}'_i)^2 \,\mathrm{d}S + \int\limits_{x \in \mathcal{S}'_i} d(x, \mathcal{S}_i)^2 \,\mathrm{d}S \right)}$$

The maximum distance of a set $\mathcal{S}$ to corresponding targets $\mathcal{S}'$ is

$$d_{max}(\mathcal{S}, \mathcal{S}') = \max_i \left(\max_{x \in \mathcal{S}_i} d(x, \mathcal{S}'_i), \max_{x \in \mathcal{S}'_i} d(x, \mathcal{S}_i) \right)$$

6.2.2 Measuring Similarity of Image Segmentations

Image segmentation assigns a label to each voxel that identifies the voxel to belong to a particular object or to its exterior otherwise. Here, a segmentation X is the set of voxels that belong to a particular object of interest.

The similarity of two segmentations X and Y can be measured by the Dice coefficient

$$\mathrm{dice}(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}$$

with the cardinality of a set $|\cdot|$. A value of 0 indicates complete mismatch while a value of 1 denotes a perfect match.

Considering multiple objects, arithmetical averages of their individual coefficients approximate an overall figure for the similarity of object compounds.

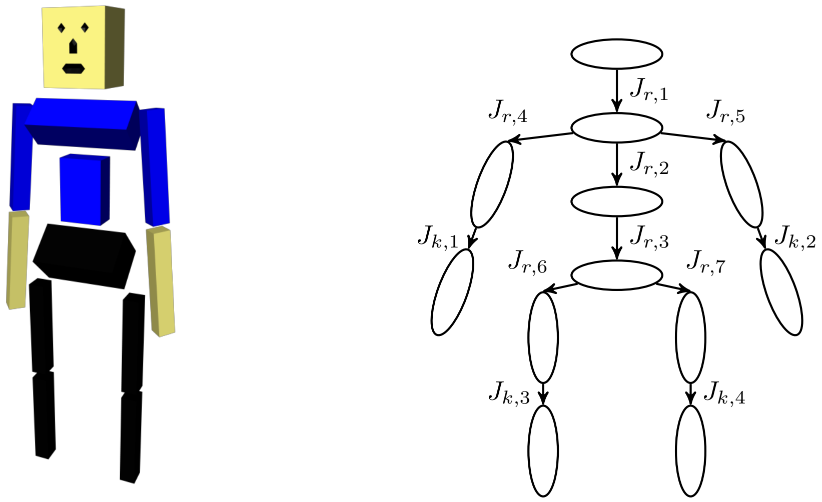
6.2.3 Measuring Invalid Overlap of Segmentations

To quantify anatomically impossible overlaps of bones during segmentation, like the one shown in Figure 1.2b, the volume of the overlap is measured. To this end, the individual objects of one segmentation are tested pairwise for intersections. In positive cases the volume of the intersection is calculated to measure the severity of the intersection.

6.3 Boxman

The Boxman is an ASSM of an artificial, humanoid-like creature with an average height of 1.7 m. An example is depicted in Figure 6.5a. Its purpose is to illustrate the capabilities of the presented approach – especially considering compounds that contain more joints than the model of the human lower limb.

The Boxman's anatomy consists of 12 cuboids. They are articulated as shown in Figure 6.5b. The spheroidal joints J_r are realized by the model presented in Section 6.1.1. The hinge joints J_k are implemented by the model of the knee that is described in Section 6.1.2.



(a) (b)

Figure 6.5: The toy example Boxman (a) and its object-joint-connectivity graph (b) comprising spheroidal joints (J_r) and hinge joints (J_k)

A spheroidal joint features three degrees of freedom, hinge joints possess two degrees of freedom each. Thus the model has a set of 29 joint posture parameters plus seven parameters determining the compound's position, orientation and size relative to its environment, making a total of 36 degrees of freedom related to pose. Variations of these parameters are illustrated in Figure 6.6.

The ASSM is trained from three differently shaped instances. These training shapes are shown in Figure 6.7.

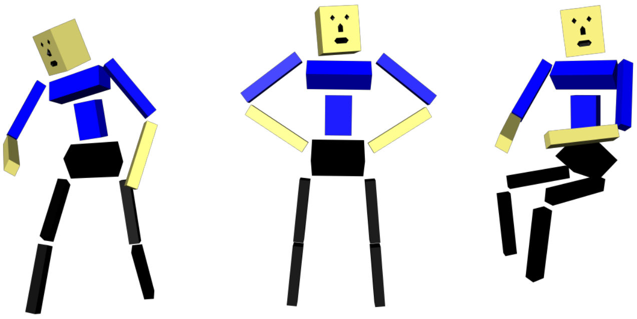


Figure 6.6: Boxman ASSM in varying poses

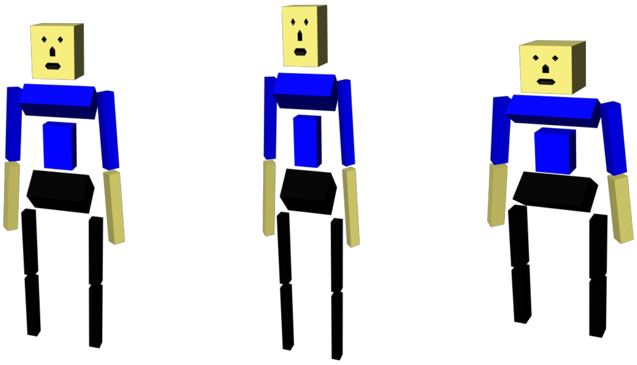


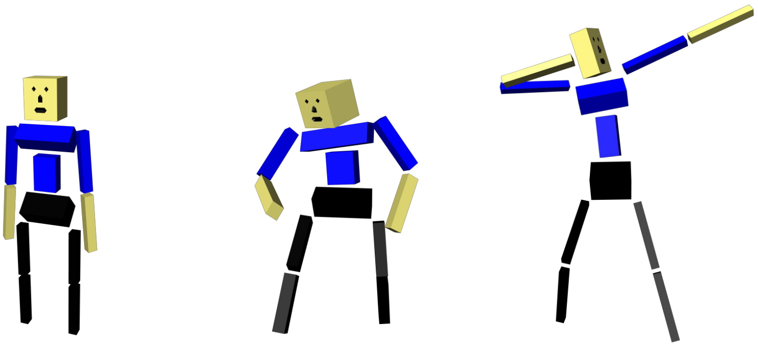
Figure 6.7: Training shapes for the Boxman ASSM

6.3.1 Experiment 1: Fitting Pose and Shape by Exploiting Correspondence

This experiment evaluates the theoretical capability of the method presented in Section 4.3.3. To this end, three Boxman target instances are generated that differ in shape and joint posture from their default configuration. A Boxman ASSM is fitted to each target by alternating optimization of pose and shape. The adaption exploits the correspondence of the ASSM's vertices to the respective target points.

Figure 6.8a shows the initial state of the ASSM that is being adapted. Figures 6.8b–d depict the targets for this experiment.

This evaluation has been conducted on a conventional PC featuring a quad-core processor at 2.8 GHz. Table 6.1 presents the results of the three adaptions. The residual of the fit is measured in terms of the distance between the ASSM's surfaces and the corresponding target surfaces. Note that the target shapes are element of the ASSM's space of allowable shapes. That is, a perfect fit is theoretically possible.



(a) Initial state (b) Target 1 (c) Target 2 (d) Target 3

Figure 6.8: Experiment setup: Initial state (a) of the ASSM that is adjusted to the targets (b–d)

Target	Residual distance to target (mm)			Iterations	Duration (s)
	d_{mean}	d_{rms}	d_{max}		
1	$18.0 \cdot 10^{-5}$	$23.2 \cdot 10^{-5}$	$81.6 \cdot 10^{-5}$	28	32
2	$4.9 \cdot 10^{-5}$	$6.4 \cdot 10^{-5}$	$24.4 \cdot 10^{-5}$	34	35
3	$6.3 \cdot 10^{-5}$	$7.7 \cdot 10^{-5}$	$24.7 \cdot 10^{-5}$	43	47

Table 6.1: Results of experiment 1

The results show that a Boxman ASSM can be fitted to a target with a maximal residual distance of approximately 10^{-4} mm. Considering that the target has a height similar to that of an adult human ($\sim 10^3$ mm), this figure proves that the fit is performed accurately. In particular, this experiment shows that the alternating optimization of shape and pose compensates for the simplifying assumption made in Section 4.3.3: Joint axes and landmarks can be considered constant under variation of shape as long as a subsequent pose adaption corrects slight pose deviations introduced by this simplification.

6.3.2 Experiment 2: Segmentation of Image Data

With this experiment the presented approach is evaluated regarding its theoretical suitability to reconstruct a geometry from a regular scalar field. To this end, Algorithm 4.3 is employed to segment binary images that depict Boxman instances in a distinct fashion, as shown in Figure 6.9. These images are generated from a manually configured ASSM. Their isotropic voxels possess an edge length of 3 mm.

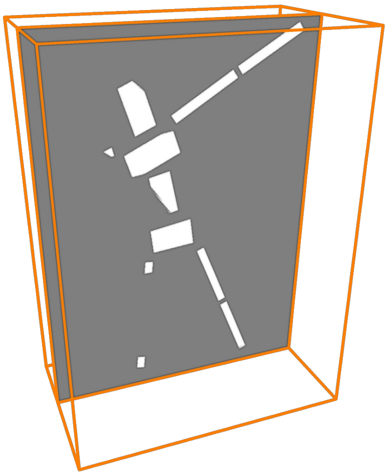
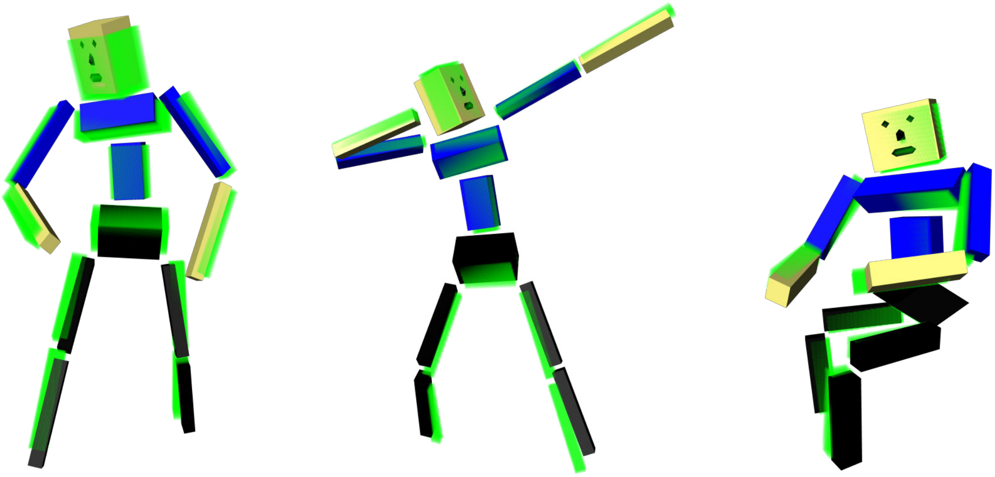


Figure 6.9: Volume of a Boxman binary image and a 2D image slice: White voxels belong to Boxman, gray voxels mark the environment

In the course of this experiment three images were segmented. They reflect the Boxman instances that are represented by the green volumes shown in Figure 6.10. The initial model configurations prior to adaption are shown as typically colored Boxmen.



(a) Case 1

(b) Case 2

(c) Case 3

Figure 6.10: Boxman images to segment (green, transparent volumes) and ASSMs (typically colored boxes) manually initialized to similar but different pose and shape that are to be fitted to the images

To adjust the ASSM, new vertex positions are determined by sampling the image on linear profiles along the current model's vertex normals. A threshold value of 0.5

on the profile denotes the estimated object boundary. Its position is the target for the respective vertex during the subsequent model adaption. The process of finding new vertex targets is illustrated in Appendix B.2.

The segmentation process consists of three stages with decreasing profile lengths and increasing sample point density. During the final stage profiles have a length of 5 cm and the distance between sample points is 1 mm.

The evaluation was carried out on the same hardware as the previous experiment. Its results are presented in Table 6.2. The residual of the fit is again measured in terms of distance to the surfaces that were used for image generation. Additionally, the match between target volume and the segmented Boxman volume is expressed by means of the Dice coefficient.

Case	Residual distance to target (mm)			Dice	Duration
	d_{mean}	d_{rms}	d_{max}		
1	0.088	0.114	0.440	0.976	5 m 57 s
2	0.010	0.013	0.102	0.996	7 m 03 s
3	0.099	0.137	0.621	0.974	5 m 19 s

Table 6.2: Results of experiment 2

The results reveal residual surface distances that are considerably smaller than the edge length of the image voxels (3 mm) for all cases. This outcome shows that an ASSM is a suitable model for the task of 3D image segmentation. It also indicates that even comprehensive models with several objects and joints exhibit a good performance.

6.4 Segmentation of the Human Lower Limb

This section describes a study to evaluate the presented method for the segmentation of medical image data. To this end, numerous ASSMs were built and employed to segment clinical computed tomography images.

The models of the human lower limb comprise the pelvis, the thighbone (femur) and the shinbone (tibia) as shown in Figure 6.11. The bones are connected by two joints: The hip and the knee. The hip, being a spheroidal joint, is modeled according to the description in Section 6.1.1. For the knee, the model presented in Section 6.1.2 is used.

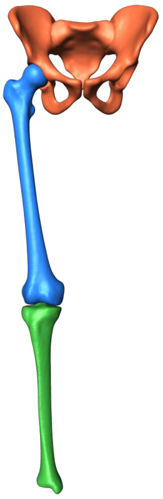


Figure 6.11: Model of the human lower limb consisting of pelvis (red), femur (blue) and tibia (green)

6.4.1 Preparation of Training Data

For the training of the models, 18 tomographic data sets of the lower limb were collected (13 CT and 5 MR images). The CT transversal images have an in-plane voxel edge length between 0.637 mm and 0.781 mm and a slice distance of 5 mm, except for two images with distances of 2.5 mm and 10 mm, respectively. Figure 6.12 shows exemplary slices of the CT data sets.

The MR images possess a voxel size of 1.56 mm × 1.56 mm in the coronal plane. In

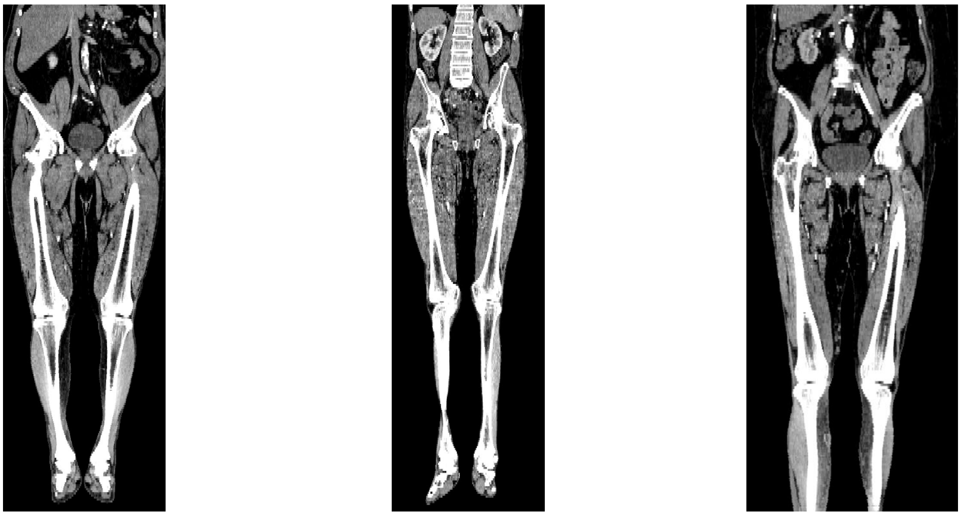

(a) (b) (c)

Figure 6.12: Exemplary slices of some CT images from which the training shapes are derived

anterior/posterior direction, the distance between slices varies between 2.5 mm and 3.3 mm.

To extract the shape of the bones of interest, the images were segmented employing automatic methods comprising SSM-based and free-form algorithms. The free-form step, which is described in [SKHLZH08], was applied to further refine the outcome. Eventually, the resulting label fields were manually reworked and corrected to represent the actual anatomies as good as possible.

Four of the CT images depict the tibia only partially, as exemplarily shown in Figure 6.12c. To still include these images in the set of training shapes, the missing parts had to be estimated from the present anatomy. To this end, an SSM of the whole tibia was fitted to the partial segmentations, resulting in complete tibiae.

The final results of the segmentation and estimation process are surface meshes that reproduce the respective anatomy of pelvis, femur and tibia.

To perform a PCA on the vertices of all data sets, the surfaces were remeshed for correspondence and aligned to a common frame of reference according to Section 4.2.

Finally, landmark points that define joint axes had to be determined. For the hip a sphere was fitted to the femoral head as shown in Figure 6.13. Its center marks the center of rotation of the hip joint. The knee landmarks were placed manually according to advice from an experienced member of the Medical Planning research group.

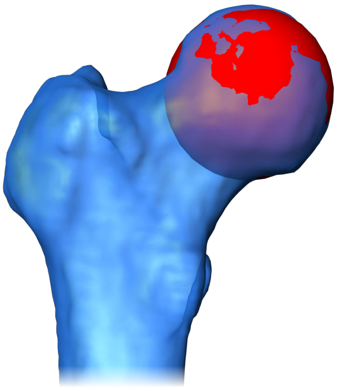


Figure 6.13: Center of rotation of the hip joint is derived from the center of a sphere fitted to the femoral head

6.4.2 The Lower Limb ASSM

The intermediate SSM of the lower limb was built by performing PCA on the set of prepared training shapes. This SSM was extended to an ASSM by the articulation model comprising joint definitions for hip and knee and the object-joint-configuration depicted in Figure 6.14.

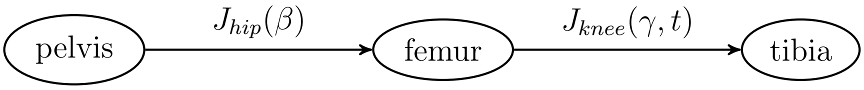


Figure 6.14: Object-joint-connectivity graph of the lower limb

Figures 6.15 and 6.16 exemplify the model's degrees of freedom with respect to joint posture and shape variation.

6.4.3 Segmentation Study

In the course of this study the 13 CT images were segmented with the help of ASSMs of the lower limb as well as by application of individual SSMs of pelvis, femur and tibia, respectively.

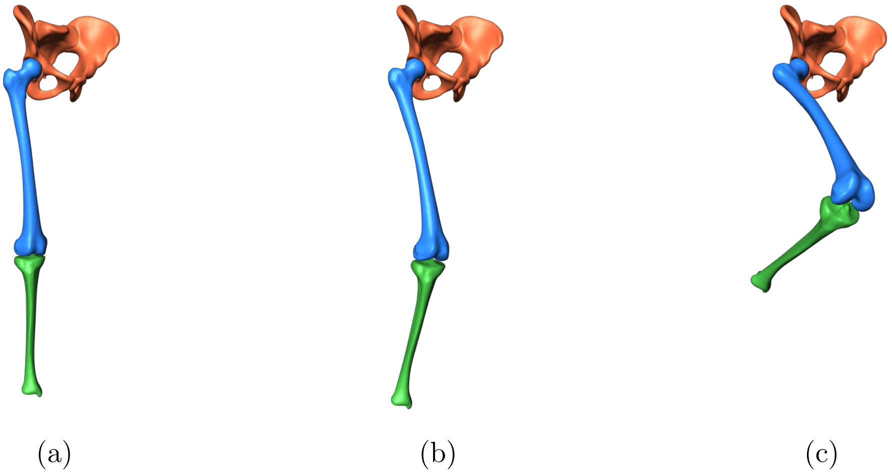

(a) (b) (c)

Figure 6.15: The lower limb ASSM with default (a) and varying joint posture (b, c)

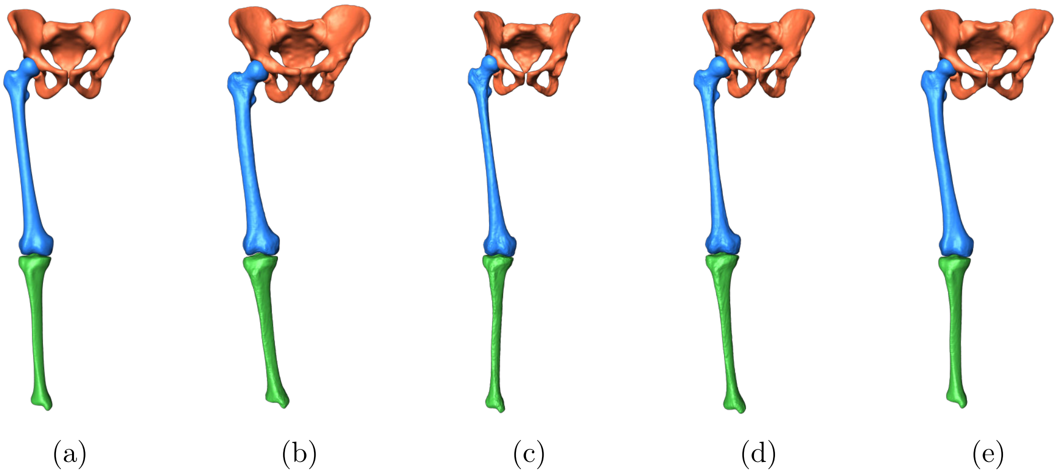

(a) (b) (c) (d) (e)

Figure 6.16: The lower limb ASSM with default (a) and varied shape (b–e)

To realize a leave-one-out cross validation, the anatomy depicted in an image was not included in the model which was employed to segment the respective image.

The approaches based on SSMs and ASSMs differ in their respective degrees of freedom of shape and pose variation and especially in the number thereof. To consider the effect of this difference, segmentation of each image was performed with SSMs and ASSMs that vary in the cardinality of their training sets and hence in the number of degrees of freedom.

To reduce any bias with respect to the underlying training set, the experiments were repeated with different training sets that are subsets of all available training shapes for a particular image.

In summary, an experiment configuration (k, l, m) was used to segment image k with the help of models that were built from l training shapes, with $1 \leq k \leq 13$ and $2 \leq l \leq 17$. m with $1 \leq m \leq 10$ represents a trial number and distinguishes different, randomly composed training sets with an exception for $l = 17$ where only one combination of training shapes is possible. For each configuration one ASSM and three SSMs were built and applied to segment according image.

Prior to segmentation the rough positions of the three bones were determined by the Generalized Hough Transform which is described in Section 3.5.1. The SSMs were initially transformed according to the results of the GHT. For the ASSM, a fit to accordingly positioned, corresponding surfaces was performed as proposed in Section 4.4.1.

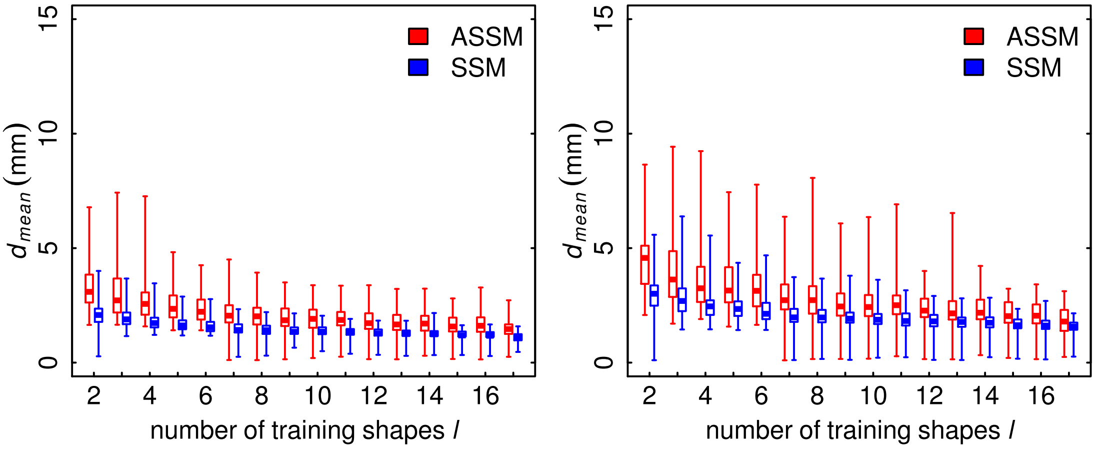
The adaption of the models to the image was performed according to Section 4.4.2. The analysis of the image in the neighborhood of the model was carried out in a way similar to the Boxman experiment (see Appendix B.2). But in this case, clinical CT images are considered. These comprise radiodensity values and may exhibit unclear information and distorting image artifacts. Hence, a more elaborate profile analysis strategy is employed, namely the algorithm presented in [SKHLZH08]. The adaption was divided into three successive stages with decreasing profile length (50 mm, 20 mm and 10 mm) and decreasing sample point distance (1 mm to 0.5 mm). Along profiles, points with a radiodensity value between 120 and 320 Hounsfield units (HU) and a gradient greater than 50 HU/mm were considered as target positions for respective vertices.

After segmentation of a data set the results stemming from ASSM and SSM adaption were compared to the reference segmentation, that is, the manually corrected segmentation the models were built from. The results of this comparison are presented in the following section.

6.4.4 Results

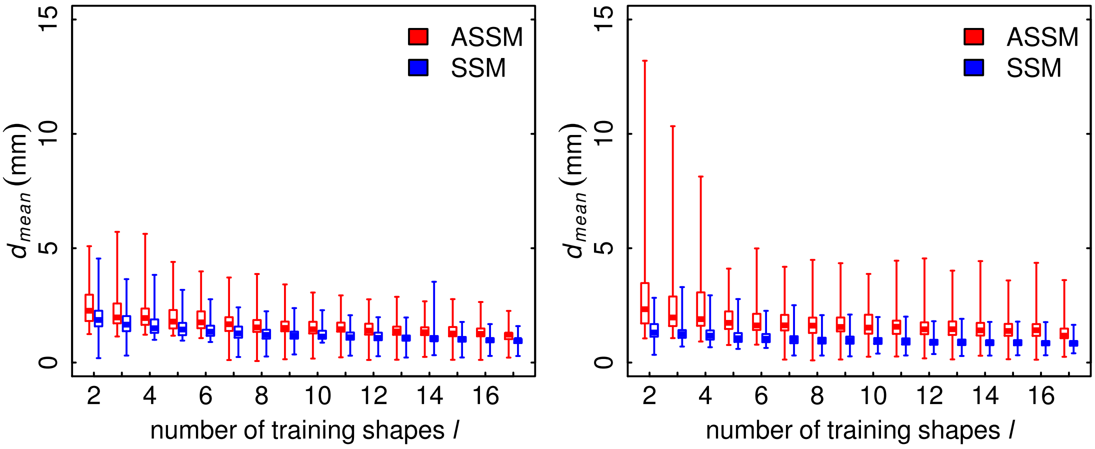
Overall Segmentation Accuracy

The overall accuracy of ASSM- and SSM-based segmentation is evaluated in terms of the surface distance to and Dice coefficient overlap with the reference segmentation. Figure 6.17 plots mean surface distance for ASSM- and SSM-based segmentations over the number of training surfaces l the models are based on. The individual values are aggregated over the image index k and the trial number m. They are shown as usual box-whisker-plots comprising minimum, lower quartile, median, upper quartile and maximum. The first chart (Figure 6.17a) presents the figures for the whole compound while the remaining charts state the values for the individual bones.



(a) Compound

(b) Pelvis



(c) Femur

(d) Tibia

Figure 6.17: Overall segmentation accuracy of an ASSM (red) compared to that of individual SSMs (blue) in terms of mean surface distance to the reference segmentation

One can see that the SSM-based segmentation method performs slightly better than the ASSM approach. However, with an increasing number of training sets l both results tend to converge. The most comprehensive models ($l = 17$) approximate the target anatomy with a median residual distance over the whole compound of

1.46 mm for ASSMs in comparison to 1.13 mm for individual SSMs.

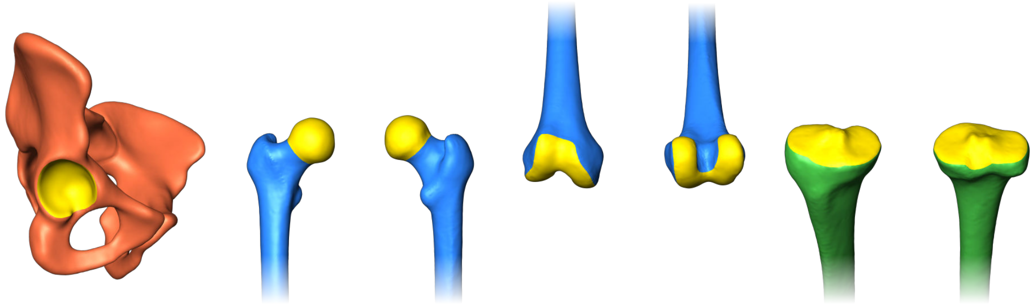
Furthermore, the results show that the estimation of the pelvic bones is less accurate than the approximation of femur and tibia. For $l = 17$ the average mean distance of the ASSM-based pelvis is 1.79 mm and the SSM pelvis' distance is 1.57 mm.

The median maximal distances to the reference are 17.51 mm for ASSMs and 14.79 mm for SSMs, as shown in Figure C.2. Analogous plots for the root-mean-square average distances d_{rms} and the Dice coefficient are also presented in Appendix C.

However, these figures are not discussed here in detail since they do not provide information beyond that of the mean surface distance for this particular study. The larger mean surface distance of the ASSM result is reflected by larger root-mean-square averaged and maximal distance values. The Dice coefficient – in which values converging to one indicate better matches – is consistent with the presented outcome as it is slightly smaller for ASSM segmentations than for the SSM results.

Segmentation Accuracy in Joint Regions

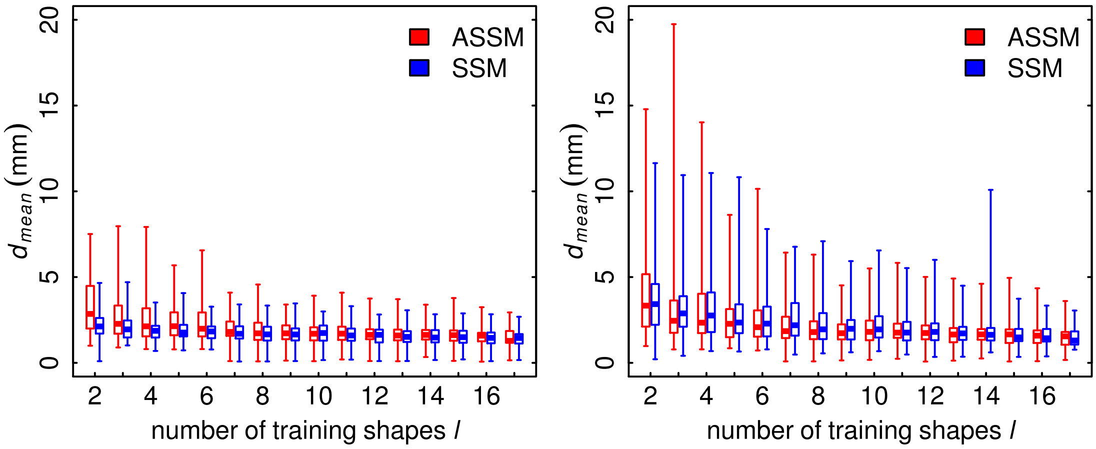
Regarding the accuracy of segmentation in the regions of joints, the mean surface distance is calculated on according subareas where adjoining bones contact each other. These patches are depicted in Figure 6.18. For the hip joint, the acetabulum (Figure 6.18a) and the femoral head (Figure 6.18b) are considered. The accuracy of knee joint segmentation is evaluated on the femoral condyles (Figure 6.18c) and the tibial plateau (Figure 6.18d).



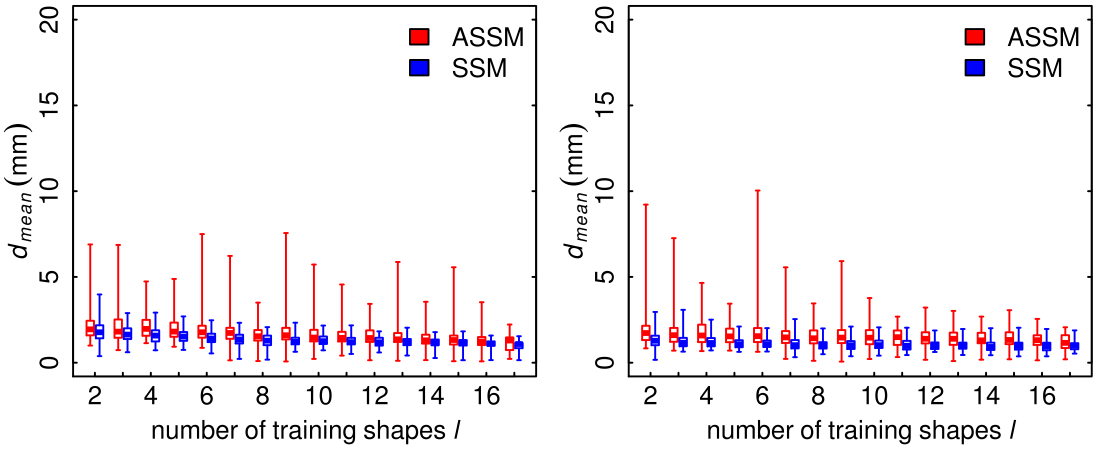
(a) Acetabulum (b) Femoral head (c) Femoral condyles (d) Tibial plateau

Figure 6.18: Yellow patches identify areas that are measured to evaluate joint segmentation accuracy

The mean surface distance of these patches to the reference is presented in Figure 6.19 analogously to the figures calculated on the entire bones.



(a) Acetabulum (b) Femoral head



(c) Femoral condyles (d) Tibial plateau

Figure 6.19: Joint segmentation accuracy of an ASSM (red) compared to that of individual SSMs (blue) in terms of mean surface distance to the reference segmentation

Especially in the region of the hip joint (see Figure 6.19a for the acetabulum and Figure 6.19b for the femoral head) the accuracy of both approaches is nearly the same. For $l = 17$ the residual mean distance is 1.41 mm for both approaches. Considering that the ASSM segmentation slightly lacks overall accuracy compared to SSM results, but provides comparable accuracy at the hip joint, this indicates a beneficial effect of the articulated coupling of pelvis and femur.

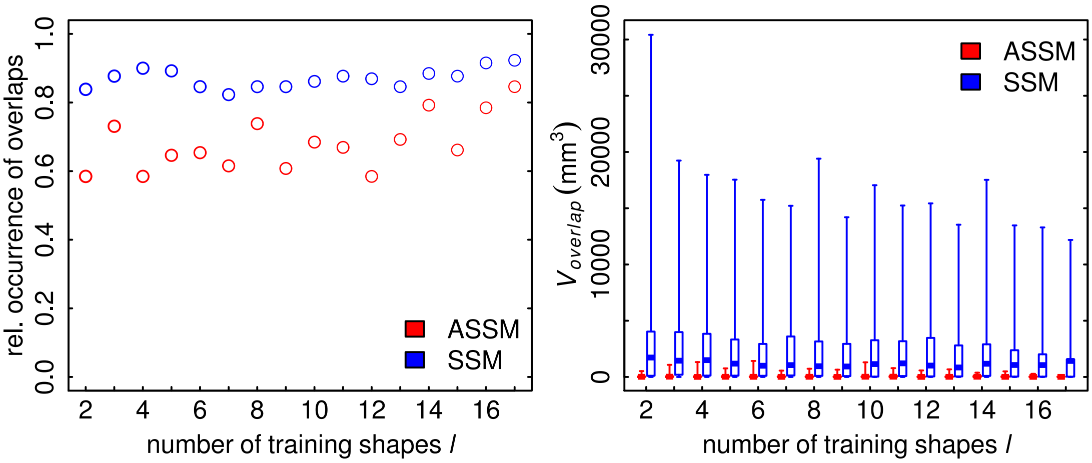
Regarding the knee region (see Figures 6.19c and 6.19d), the ASSM results ($d_{mean} =$

1.22 mm for $l = 17$) slightly lack accuracy compared to the SSM-based results ($d_{mean} = 1.02$ mm for $l = 17$). Here, the coupling of femur and tibia does not seem to be particularly beneficial to the segmentation result.

Occurrence and Severity of Invalid Overlaps

As stated in Section 1.1, a segmentation of adjoining objects by multiple, independent SSMs may be invalid and thus unusable if the individual segmentations overlap. To compare ASSMs and SSMs with respect to that concern, the potential overlaps are quantified by their rate of occurrence (ratio of overlapping segmentations to overall segmentations) and their respective volume.

Figure 6.20 presents these figures for the hip joint. One can see that overlaps occur with both, SSMs and ASSMs. However, the ASSM-based segmentations contain overlaps less often (67% of all segmentations exhibit overlaps) than the results produced by SSMs (87% overlapping segmentations). Also, the overlaps from ASSMs are small in terms of their volume (few cubic millimeters). In contrast, the overlaps from SSMs are in the range of a few cubic centimeter. These results show that the segmentation of the hip joint benefits from the use of an ASSM. However, the ASSM segmentations still contain marginal overlaps.

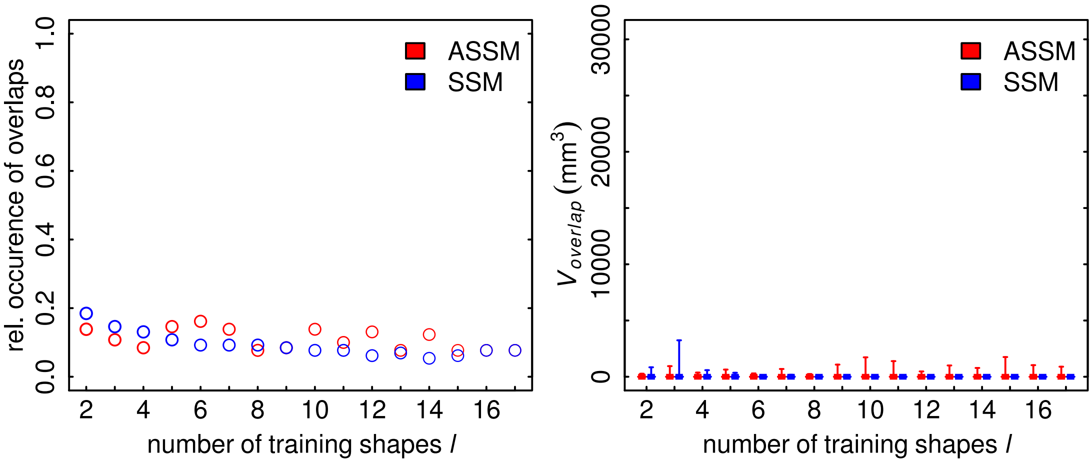


(a) Relative occurrence of overlaps

(b) Overlap volume

Figure 6.20: Occurrence and severity of invalid, overlapping segmentations at the hip

The results for the knee joint differ clearly from those for the hip. They are presented in Figure 6.21. Here, the results of both approaches, SSM- and ASSM-based, are quite similar. Invalid overlaps occur rather rarely (11% of ASSM segmentations contain overlaps, 10% of SSM segmentations overlap) and their volume in the range of cubic millimeters is comparatively small.



(a) Relative occurrence of overlaps

(b) Overlap volume

Figure 6.21: Occurrence and severity of invalid, overlapping segmentations at the knee

In contrast to the hip, the segmentation of the knee joint appears to be less problematic for both models, SSMs and ASSMs. This may be due to the thicker cartilage that separates femur and tibia. Thereby, the gap between respective bones is depicted more clearly in tomographic images and segmentation can be performed more reliably.

7 Discussion

This work proposes a generic approach to build and use articulated statistical shape models (ASSMs) for anatomy reconstruction tasks. Its capabilities are evaluated with the help of two types of ASSMs: The artificial Boxman and the model of the lower limb.

7.1 Theoretical Performance

Two experiments were conducted with the Boxman model to evaluate the theoretical performance of the proposed method.

The first experiment (see Section 6.3.1) tests the capability to estimate an articulated target shape based on corresponding target positions for each vertex of the Boxman ASSM. Its results show that the presented method performs well even for comprehensive ASSMs.

The second experiment presented in Section 6.3.2 verifies the suitability of an ASSM to be employed for the task of image segmentation. To this end, the Boxman ASSM is employed to segment volumetric binary images. The results exhibit residual distances to the target shapes that are smaller than the resolution of respective images. This shows the capability of the presented approach for segmentation tasks.

7.2 Practical Outcome

The model of the human lower limb is employed to evaluate the performance of the ASSM method in a practical medical segmentation scenario.

For the comprehensive study, a set of CT images of the human lower limb were segmented by ASSMs comprising pelvis, femur and tibia and by individual SSMs of respective bones. Section 6.4.4 presents the aggregated results over the number of data sets used to train the employed models. First of all, one can see that segmentations stemming from ASSMs are comparable, but slightly less accurate than the SSM-based results.

To discuss this outcome, one must consider the degrees of freedom, that is the configuration space, of the applied models. For this purpose, Table 7.1 compares the pose and shape parameters for the most comprehensive models (maximal number of training shapes) of the human lower limb.

	One ASSM	Three SSMs (pelvis, femur, tibia)
Shape DoFs	16	$3 \times 16 = 48$
Pose DoFs	$7 + 3 + 2 = 12$ (global pose, hip, knee)	$3 \times 7 = 21$
Total	28	69

Table 7.1: Degrees of freedom (DoFs) of a lower limb ASSM compared to those of three individual SSMs

Regarding the pose of the individual bones, the three SSMs feature a total of 21 pose degrees of freedom. They allow independent positioning and scaling of the individual objects – and permit non-natural relative poses also (see Figure 1.2a). With the ASSM approach, the relative poses are determined by five joint parameters that reflect anatomical joint postures. The global pose and size of the compound are controlled by seven additional parameters.

The shape of the SSMs is determined independently by 16 parameters per model in contrast to 16 parameters for the shape of the entire ASSM compound. Here, the SSMs allow greater flexibility regarding deformation, but also permit shape instances of adjacent objects that do not match each other (see Figure 1.2b).

Considering the present study, the flexibility of SSMs benefits their segmentation accuracy: The results from a lower limb ASSM trained from 17 training shapes ($d_{mean} = 1.46\,\text{mm}$) are approximately as accurate as results obtained from three SSMs built from 6 or 7 data sets (d_{mean} between 1.43 and 1.54 mm).

On the other hand, this flexibility leads to a larger number of invalid segmentations, especially in the region of the hip joint. Here, an ASSM produces less overlapping segmentations. Furthermore, the severity of these overlaps is drastically decreased by the articulated coupling of objects. Also, the quality of segmentation at the hip is almost the same for SSMs and ASSMs. Accounting for the reduced configuration space of ASSMs in comparison to SSMs, this indicates a beneficial effect of the articulated model.

For the knee, these effects could not be observed: Both, SSM- and ASSM-based segmentations, exhibit overlaps of femur and tibia comparatively rarely and their volume of intersection is very small. This may be due to the larger gap that separates those bones – here, conventional SSMs suffice to clearly distinguish the adjacent objects.

In conclusion, individual SSMs exhibit a greater range of deformation and positioning possibilities. Especially with a small number of training shapes, this benefits the approximation of anatomies not being part of the training set. But it also permits non-natural shape and pose. In contrast, an ASSM of the same anatomical structure is more restrictive in terms of shape and pose. Hence, to reach a comparable deformation flexibility, a more comprehensive set of training data must be used.

7.3 Improving ASSMs

As shown in Section 6.4.4, even ASSMs may exhibit slightly overlapping configurations. This may be due to

- imprecisely segmented training data,
- imprecise identification and fixation of joint axes and landmarks or
- insufficiently modeled joints.

These causes may not be avoided completely, however, there is room for improvement. Training shapes may be generated from images with higher resolution. This would improve the quality of shape and would also facilitate the correct identification of anatomical axes and landmarks.

Addressing the last point, the presented articulation model clearly depends on good joint models, that is, models that represent the full spectrum of natural posture, but prohibit non-natural configurations. However, real joints exhibit complex mechanisms of motion and the determination of their functional axes is subject to ongoing research. For this reason, joint models should be implemented with great care and under consideration of recent knowledge.

7.4 An Alternative Idea for ASSMs

As already discussed, ASSMs as presented here require an increased number of training shapes to reach the shape variability of conventional SSMs. However, it is feasible to build articulated models that can compete with SSMs regarding that concern. To this end, one may apply the presented articulation model (see Section 4.1) to individual SSMs rather than to one SSM of an object compound. That is, the objects of the compound may be deformed independently – sacrificing confidence in matching shape of adjacent objects.

8 Conclusion and Future Work

This work presents a framework to build articulated statistical shape models and employ them for anatomy reconstruction tasks. The approach is capable to support a broad range of joints, which can be designed in a straightforward manner. Also an implementation is described that facilitates the seamless replacement of conventional statistical shape models with articulated models.

The study conducted in the course of this work shows that ASSMs can be applied to image segmentation tasks. The configuration space of those models better reflects the natural degrees of freedom of articulated anatomical structures than conventional SSMs. In the present study, this particularly benefits the reconstruction of the hip joint. The number and severity of obviously invalid segmentations can be reduced by application of the new method.

However, the construction of such models is more labor intensive than that of comparable SSMs. ASSMs require a more comprehensive set of training shapes to reach the accuracy of conventional SSMs. Furthermore, the required joint models must be designed first. Since the anatomy of real joints is very complex, this is indeed a challenging task.

Future efforts to enhance the presented approach should be targeted at the design of those joint models. It may be feasible to develop models that explicitly consider the validity of particular poses by observing and prohibiting overlaps. Additionally, small displacements of an object relative to an adjoint one could be introduced to relax the potentially over-restrictive constraints of the present models. In that respect, the author foresees a great benefit from a collaboration with research groups that concentrate on the biomechanical analysis of joint anatomies, e.g. the Computational Medicine group of the Zuse-Institute Berlin.

Also, new ASSMs should be implemented to verify the present approach with respect to other anatomical structures. Especially ASSMs that are based upon a more comprehensive set of training shapes have to prove their advantages over individual SSMs.

A Auxiliary Equations and Rationale

A.1 Conversion between Euclidean and Homogeneous Coordinates

Converting Euclidean to homogeneous coordinates:

$$(x, y, z)^T \mapsto (x, y, z, 1)^T$$

Converting homogeneous to Euclidean coordinates:

$$(x, y, z, w)^T \mapsto (x/w, y/w, z/w)^T$$

A.2 Invertibility of Transformations

A transformation can be inverted by the inversion of its matrix. The transformations considered in this work are translation, rotation, uniform scaling and compositions thereof.

A translation is represented by a 4×4 identity matrix with the last column being the translation vector. That is, its rows are linearly independent. Hence, such a matrix is invertible.

Rotations are expressed by orthogonal matrices which are invertible.

Scaling matrices possess only diagonal elements that are non-null. Prohibiting scaling by zero, these matrices are invertible.

Also, the result of a concatenation of invertible matrices is an invertible matrix. Thus, all transformation and their respective matrix representations considered in this work are invertible.

A.3 Basic Transformations and their Elementwise Partial Derivatives

Translation by vector $\vec{v}$:

$$\operatorname{transl}(\vec{v}) = \begin{pmatrix} 1 & 0 & 0 & v_1 \\ 0 & 1 & 0 & v_2 \\ 0 & 0 & 1 & v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \frac{\partial \operatorname{transl}(\vec{v})}{\partial v_1} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial \operatorname{transl}(\vec{v})}{\partial v_2} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \frac{\partial \operatorname{transl}(\vec{v})}{\partial v_3} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Translation by $t \in \mathbb{R}$ in direction of fixed vector $\vec{v}$:

$$\operatorname{transl}_{\vec{v}}(t) = \begin{pmatrix} 1 & 0 & 0 & tv_1 \\ 0 & 1 & 0 & tv_2 \\ 0 & 0 & 1 & tv_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \frac{\partial \operatorname{transl}_{\vec{v}}(t)}{\partial t} = \begin{pmatrix} 0 & 0 & 0 & v_1 \\ 0 & 0 & 0 & v_2 \\ 0 & 0 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Rotations around coordinate axes:

$$\mathrm{rot}_x(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \frac{\partial\,\mathrm{rot}_x(\phi)}{\partial\phi} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -\sin\phi & -\cos\phi & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathrm{rot}_y(\phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \frac{\partial\,\mathrm{rot}_y(\phi)}{\partial\phi} = \begin{pmatrix} -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 0 \\ -\cos\phi & 0 & -\sin\phi & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathrm{rot}_z(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \frac{\partial\,\mathrm{rot}_z(\phi)}{\partial\phi} = \begin{pmatrix} -\sin\phi & -\cos\phi & 0 & 0 \\ \cos\phi & -\sin\phi & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Rotation around a fixed unit axis $\vec{v}$:

$$\mathrm{rot}_{\vec{v}}(\phi) = \begin{pmatrix} \cos\phi + v_1^2 \operatorname{ver}\phi & v_1 v_2 \operatorname{ver}\phi - v_3 \sin\phi & v_1 v_3 \operatorname{ver}\phi + v_2 \sin\phi & 0 \\ v_1 v_2 \operatorname{ver}\phi + v_3 \sin\phi & \cos\phi + v_2^2 \operatorname{ver}\phi & v_2 v_3 \operatorname{ver}\phi - v_1 \sin\phi & 0 \\ v_1 v_3 \operatorname{ver}\phi - v_2 \sin\phi & v_2 v_3 \operatorname{ver}\phi + v_1 \sin\phi & \cos\phi + v_3^2 \operatorname{ver}\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{with } \operatorname{ver}\phi = 1 - \cos\phi$$

$$\frac{\partial\,\mathrm{rot}_{\vec{v}}(\phi)}{\partial\phi} = \begin{pmatrix} -\sin\phi + v_1^2 \sin\phi & v_1 v_2 \sin\phi - v_3 \cos\phi & v_1 v_3 \sin\phi + v_2 \cos\phi & 0 \\ v_1 v_2 \sin\phi + v_3 \cos\phi & -\sin\phi + v_2^2 \sin\phi & v_2 v_3 \sin\phi - v_1 \cos\phi & 0 \\ v_1 v_3 \sin\phi - v_2 \cos\phi & v_2 v_3 \sin\phi + v_1 \cos\phi & -\sin\phi + v_3^2 \sin\phi & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Uniform scaling by scalar s:

$$\mathrm{scale}(s) = \begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \frac{\partial\, \mathrm{scale}(s)}{\partial s} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

A.4 Gradient of the Alignment Objective Function: Auxiliary Calculation

$$\begin{aligned}
v^T \left(\frac{\partial}{\partial \alpha_s} T^T T \right) v &= \sum_{klm} v^{(k)} \left(\frac{\partial}{\partial \alpha_s} T^{(lk)} T^{(lm)} \right) v^{(m)} \\
&= \sum_{klm} \left(v^{(k)} \frac{\partial T^{(lk)}}{\partial \alpha_s} T^{(lm)} v^{(m)} + v^{(k)} T^{(lk)} \frac{\partial T^{(lm)}}{\partial \alpha_s} v^{(m)} \right) \\
&= \sum_{klm} v^{(k)} \frac{\partial T^{(lk)}}{\partial \alpha_s} T^{(lm)} v^{(m)} + \sum_{klm} v^{(k)} T^{(lk)} \frac{\partial T^{(lm)}}{\partial \alpha_s} v^{(m)} \\
&= \sum_{mlk} v^{(m)} \frac{\partial T^{(lm)}}{\partial \alpha_s} T^{(lk)} v^{(k)} + \sum_{klm} v^{(k)} T^{(lk)} \frac{\partial T^{(lm)}}{\partial \alpha_s} v^{(m)} \\
&= 2 \cdot \sum_{klm} v^{(k)} T^{(lk)} \frac{\partial T^{(lm)}}{\partial \alpha_s} v^{(m)} \\
v^T \left(\frac{\partial}{\partial \alpha_s} T^T T \right) v &= 2 \cdot v^T T^T \frac{\partial T}{\partial \alpha_s} v
\end{aligned} \tag{A.1}$$

A.5 Efficient Calculation of the Alignment Objective Function Value

$$\begin{aligned}
d(\alpha) &= \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left| T_i(\alpha)\, v_{ij} - u_{ij} \right|^2 w_{ij} \\
&= \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left(T_i(\alpha)\, v_{ij} - u_{ij} \right)^T \left(T_i(\alpha)\, v_{ij} - u_{ij} \right) w_{ij}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{M}\sum_{j=1}^{N_i} \left(v_{ij}^T T_i^T T_i v_{ij} - u_{ij}^T T_i v_{ij} - v_{ij}^T T_i^T u_{ij} + u_{ij}^T u_{ij}\right) w_{ij} \\
&= \sum_{i=1}^{M}\sum_{j=1}^{N_i} \left(v_{ij}^T T_i^T T_i v_{ij} - 2u_{ij}^T T_i v_{ij} + u_{ij}^T u_{ij}\right) w_{ij} \\
&= \sum_{i=1}^{M}\sum_{j=1}^{N_i}\sum_{klm} v_{ij}^{(k)} T_i^{(lk)} T_i^{(lm)} v_{ij}^{(m)} w_{ij} - 2\sum_{i=1}^{M}\sum_{j=1}^{N_i}\sum_{kl} u_{ij}^{(k)} T_i^{(kl)} v_{ij}^{(l)} w_{ij} \\
&\qquad + \underbrace{\sum_{i=1}^{M}\sum_{j=1}^{N_i}\sum_{k} u_{ij}^{(k)} u_{ij}^{(k)} w_{ij}}_{=U} \\
&= \sum_{i=1}^{M}\sum_{klm}\sum_{j=1}^{N_i} v_{ij}^{(k)} T_i^{(lk)} T_i^{(lm)} v_{ij}^{(m)} w_{ij} - 2\sum_{i=1}^{M}\sum_{kl}\sum_{j=1}^{N_i} u_{ij}^{(k)} T_i^{(kl)} v_{ij}^{(l)} w_{ij} + U \\
&= \sum_{i=1}^{M}\sum_{klm} T_i^{(lk)} T_i^{(lm)} \underbrace{\sum_{j=1}^{N_i} v_{ij}^{(k)} v_{ij}^{(m)} w_{ij}}_{=V_i^{(km)}} - 2\sum_{i=1}^{M}\sum_{kl} T_i^{(kl)} \underbrace{\sum_{j=1}^{N_i} u_{ij}^{(k)} v_{ij}^{(l)} w_{ij}}_{=\mathcal{U}_i^{(kl)}} + U
\end{aligned}
$$

$$
d(\alpha) = \sum_{i=1}^{M}\sum_{klm} T_i^{(lk)}(\alpha)\, T_i^{(lm)}(\alpha)\, V_i^{(km)} - 2\sum_{i=1}^{M}\sum_{kl} T_i^{(kl)}(\alpha)\, \mathcal{U}_i^{(kl)} + U \tag{A.2}
$$

For an efficient repeated evaluation of d w.r.t. different parameters α it is beneficial to compute the constants U, V_i and $\mathcal{U}_i$ once and store them for substitution. This way, the expensive loops over the objects' vertices must be executed only once.

A.6 Efficient Calculation of the Gradient of the Alignment Objective Function

The gradient of the objective functions consists of the partial derivatives w.r.t the components of α. With Equation (A.2) they can be written as

$$
\begin{aligned}
\frac{\partial d(\alpha)}{\partial \alpha_s} = \frac{\partial}{\partial \alpha_s} \Bigg(& \sum_{i=1}^{M}\sum_{klm} T_i^{(lk)}(\alpha)\, T_i^{(lm)}(\alpha)\, V_i^{(km)} \\
& - 2\sum_{i=1}^{M}\sum_{kl} T_i^{(kl)}(\alpha)\, \mathcal{U}_i^{(kl)} + U \Bigg)
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\partial}{\partial \alpha_s} \sum_{i=1}^{M} \sum_{klm} T_i^{(lk)}(\alpha)\, T_i^{(lm)}(\alpha)\, V_i^{(km)} \\
&\qquad\qquad - 2\frac{\partial}{\partial \alpha_s} \sum_{i=1}^{M} \sum_{kl} T_i^{(kl)}(\alpha)\, \mathcal{U}_i^{(kl)} + \underbrace{\frac{\partial U}{\partial \alpha_s}}_{=0} \\
&= \sum_{i=1}^{M} \sum_{klm} \left(\frac{\partial}{\partial \alpha_s} T_i^{(lk)}(\alpha)\, T_i^{(lm)}(\alpha) \right) V_i^{(km)} \\
&\qquad\qquad - 2 \sum_{i=1}^{M} \sum_{kl} \frac{\partial T_i^{(kl)}(\alpha)}{\partial \alpha_s} \mathcal{U}_i^{(kl)} \\
&= \sum_{i=1}^{M} \sum_{klm} \left(\frac{\partial T_i^{(lk)}(\alpha)}{\partial \alpha_s} T_i^{(lm)}(\alpha) + T_i^{(lk)}(\alpha) \frac{\partial T_i^{(lm)}(\alpha)}{\partial \alpha_s} \right) V_i^{(km)} \\
&\qquad\qquad - 2 \sum_{i=1}^{M} \sum_{kl} \frac{\partial T_i^{(kl)}(\alpha)}{\partial \alpha_s} \mathcal{U}_i^{(kl)}
\end{aligned}
$$

For an efficient repeated evaluation of the partial derivative of d it is beneficial to compute the constants V_i and $\mathcal{U}_i$ once and store them for substitution. Also, a simultaneous evaluation of the objective function and its gradient is computationally efficient, since both formulas consider V_i and $\mathcal{U}_i$ (see Appendix A.5).

B Examples

B.1 Determining Object Transformations

This example illustrates the algorithm presented in Section 4.1.3 on the connectivity graph depicted in Figure 4.2.

Here, the pseudo-object representing the compound's environment is attached to the second object, which is associated with its transformation T_2, as depicted in Figure B.1.

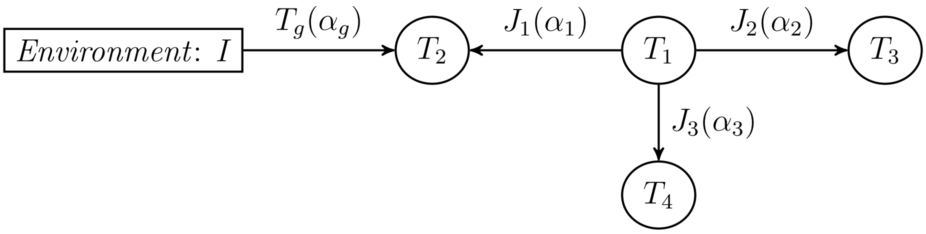


Figure B.1: Attaching the *Environment* pseudo-object to the second object

The algorithm starts with the pseudo-object *Environment* and its identity transformation and derives the pose of the second object as

$$T_2(\alpha) = I \circ T_{global}(\alpha_g) = T_{global}(\alpha_g)$$

To continue traversal the edge from T_1 to T_2 must be inverted as shown in Figure B.2.

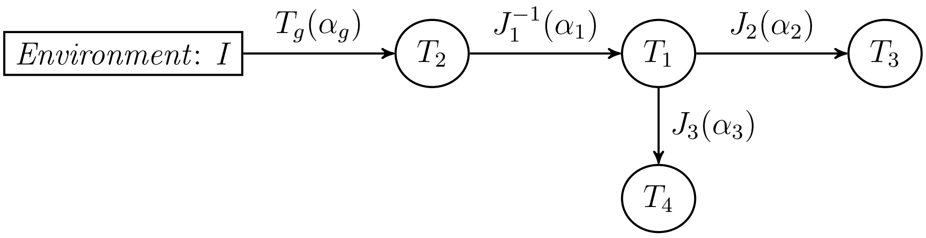


Figure B.2: A more complex compound of connected objects

Continuing via T_1 to T_3 and T_4 completes the concatenation of transformations to

$$\begin{aligned}
T_2(\alpha) &= I \circ T_{global}(\alpha_g) &&= T_{global}(\alpha_g)\\
T_1(\alpha) &= I \circ T_{global}(\alpha_g) \circ J_1^{-1}(\alpha_1) &&= T_{global}(\alpha_g) \circ J_1^{-1}(\alpha_1)\\
T_3(\alpha) &= I \circ T_{global}(\alpha_g) \circ J_1^{-1}(\alpha_1) \circ J_2(\alpha_2) &&= T_{global}(\alpha_g) \circ J_1^{-1}(\alpha_1) \circ J_2(\alpha_2)\\
T_4(\alpha) &= I \circ T_{global}(\alpha_g) \circ J_1^{-1}(\alpha_1) \circ J_3(\alpha_3) &&= T_{global}(\alpha_g) \circ J_1^{-1}(\alpha_1) \circ J_3(\alpha_3)
\end{aligned}$$

B.2 Determining Corresponding Target Positions during Image Segmentation

During ASSM-based 3D image segmentation, the ASSM is fitted to the shape that is depicted by the image. Figure B.3 shows an exemplary image and an ASSM that is to be fitted to that image. To this end, the model's shape and pose are adjusted by application of Algorithm 4.3 which requires an updated target position for each vertex of the ASSM in each iteration. This section exemplarily describes how these targets are determined with the help of Boxman (see Section 6.3).

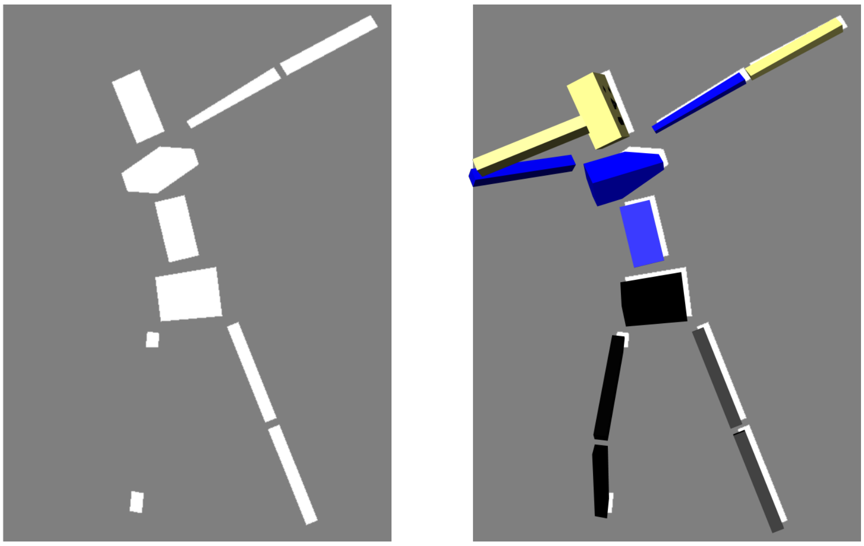
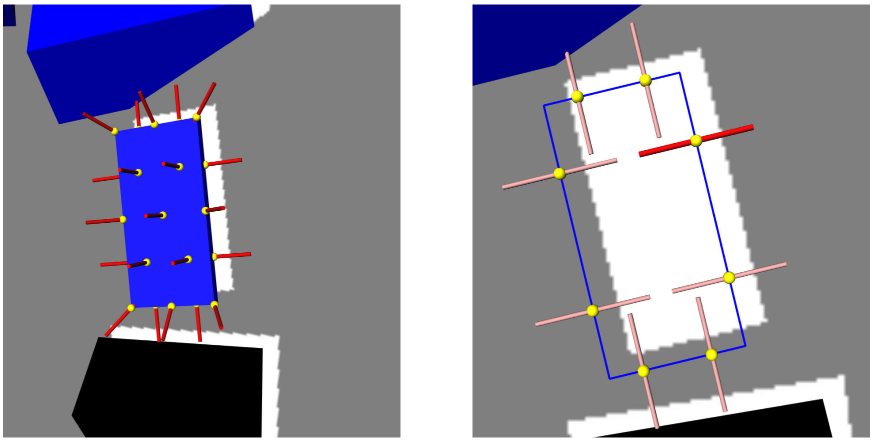

(a) Slice of a 3D Boxman image (b) ASSM during adaptation

Figure B.3: A Boxman image and the Boxman ASSM

To determine a new target position u_{ij} for a vertex v_{ij}, an intensity function is sampled along a linear profile. The profile's center is the vertex itself and it runs in direction of the surface normal vector as depicted in Figure B.4.

Now the intensity function is searched for a point where its values cross a particular threshold (Figure B.5a). Since image voxels that belong to Boxman are labeled as 1 and exterior voxels are labeled as 0, the threshold value is 0.5 here. That is, the spatial point on the profile with a value of 0.5 is the target position of respective vertex (Figure B.5b).

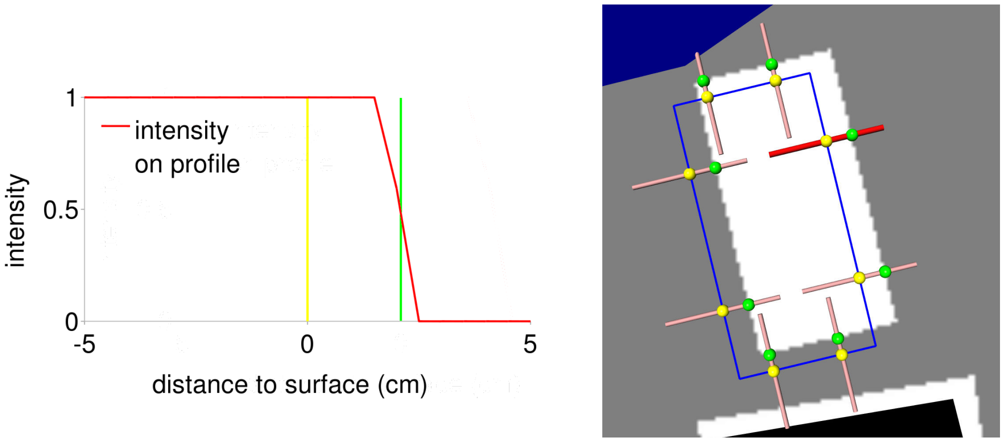
Regarding the case that such a target cannot be found due to inappropriate sampling



(a) 3D view

(b) Projection onto 2D slice

Figure B.4: Profiles (red) through ASSM vertices (yellow) in surface normal direction



(a) Image values sampled along the highlighted profile, the green position meets the threshold criterion (value of 0.5)

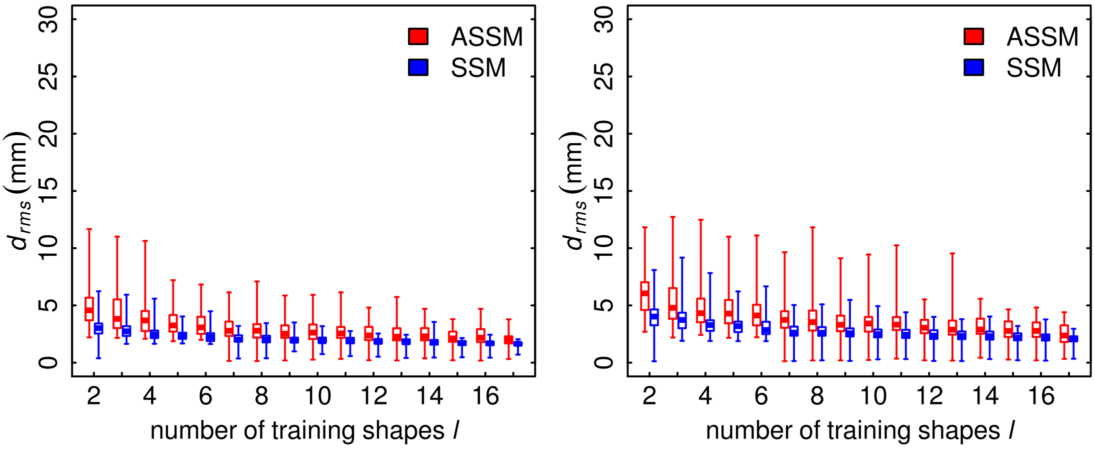
(b) Determined target positions (green) for a couple of vertices

Figure B.5: Search for target position along profiles

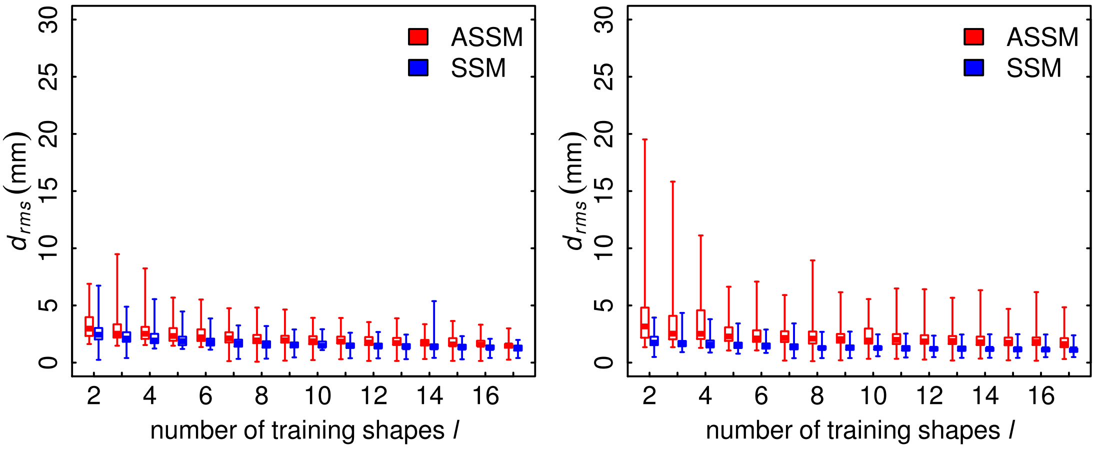
values, the weight w_{ij} of the target is set to zero. Hence, only certain targets are considered during the adaptation of the model.

The threshold value search is a rather naive method. In the course of this work, it is only employed for the theoretical Boxman experiment. To process medical images, like CT or MR images, more complex methods are used. For details, the reader is referred to [SKHLZH08].

C Additional Evaluation Figures

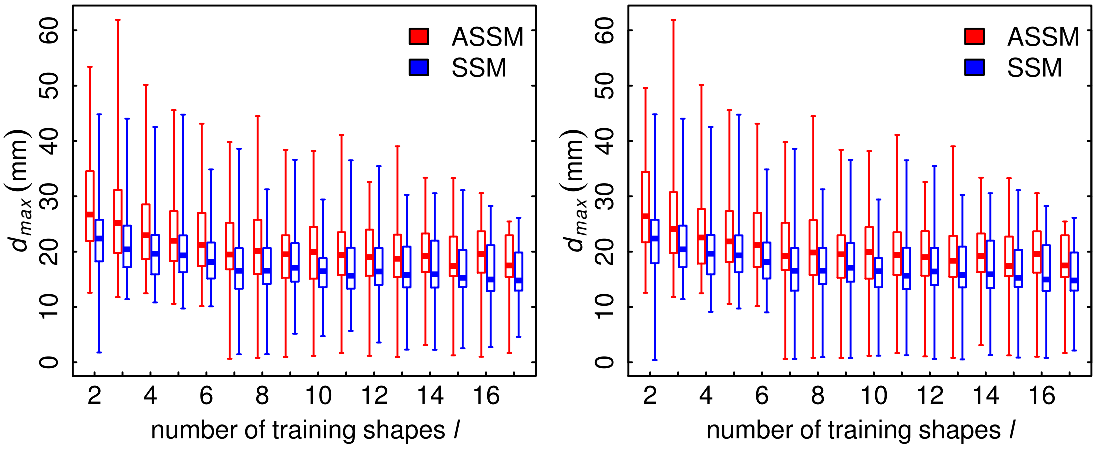


(a) Compound (b) Pelvis

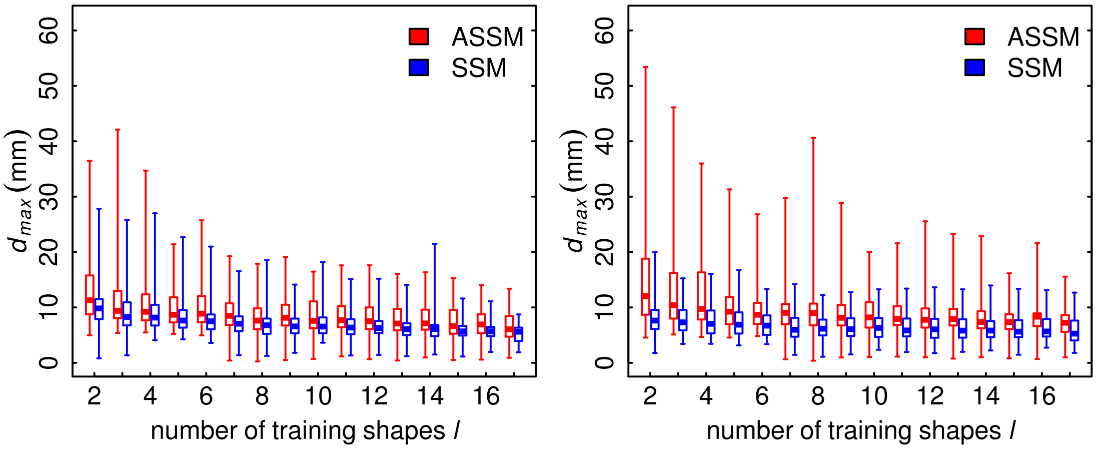


(c) Femur (d) Tibia

Figure C.1: Overall segmentation accuracy of an ASSM (red) compared to that of individual SSMs (blue) in terms of root-mean-square surface distance to the reference segmentation

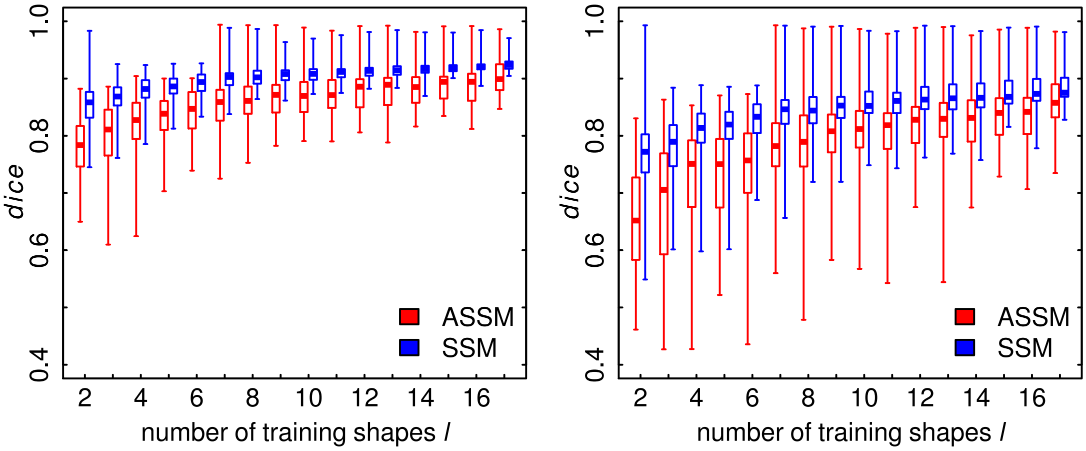


(a) Compound (b) Pelvis

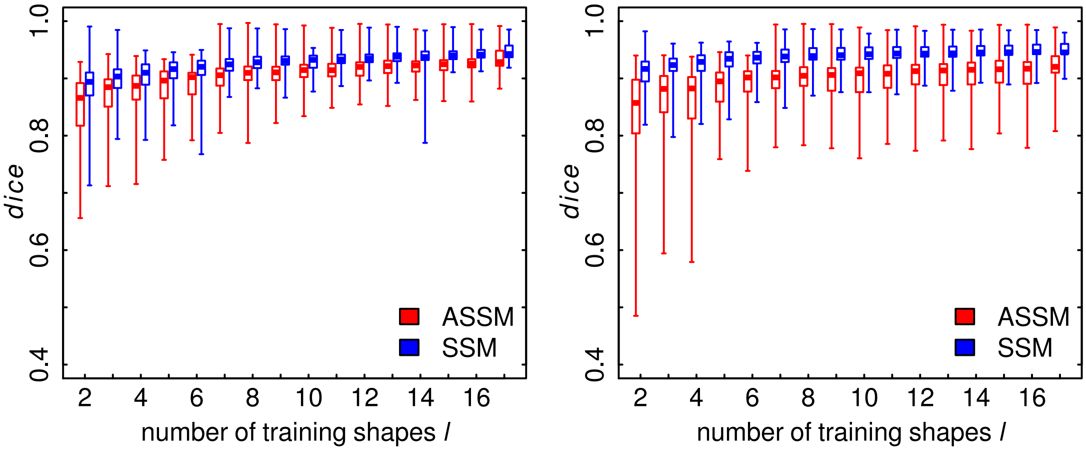


(c) Femur (d) Tibia

Figure C.2: Overall segmentation accuracy of an ASSM (red) compared to that of individual SSMs (blue) in terms of maximal surface distance to the reference segmentation



(a) Compound (b) Pelvis



(c) Femur (d) Tibia

Figure C.3: Similarity of segmentations to the reference of ASSMs (red) compared to that of individual SSMs (blue) in terms of Dice coefficient

Bibliography

[Ami] *ZIBAmira Homepage*. 2012. URL: `http://amira.zib.de/index.html`.

[Bal81] D. H. Ballard. “Generalizing the Hough Transform to Detect Arbitrary Shapes”. In: *Pattern Recognition* 13.2 (Jan. 1981), pp. 111–122. URL: `http://www.cs.utexas.edu/~dana/HoughT.pdf`.

[BKRLZ12] Matthias Bindernagel, Dagmar Kainmüller, Heiko Ramm, Hans Lamecker, and Stefan Zachow. “Analysis of inter-individual anatomical shape variations of joint structures”. In: *Proc. Int. Society of Computer Assisted Orthopaedic Surgery (CAOS)*. Vol. 210. 2012. URL: `http://www.zib.de/lamecker/publications/caos-2012b.pdf`.

[BKSLZH11] Matthias Bindernagel, Dagmar Kainmüller, Heiko Seim, Hans Lamecker, Stefan Zachow, and Hans-Christian Hege. “An Articulated Statistical Shape Model of the Human Knee”. In: *Bildverarbeitung für die Medizin 2011*. Informatik aktuell (2011). Ed. by H. Handels and et al., pp. 59–63. URL: `http://www.zib.de/lamecker/publications/bvm-2011a.pdf`.

[CD08] Carl D. Crane III and J. Duffy. *Kinematic Analysis of Robot Manipulators*. Cambridge University Press, 2008. ISBN: 9780521047937. URL: `http://books.google.de/books?id=2e4_gb4wTjIC`.

[CTCG95] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. “Active Shape Models – Their Training and Application”. In: *Computer Vision and Image Understanding* 61.1 (1995), pp. 38–59.

[DM98] Ian L. Dryden and K.V. Mardia. *Statistical Shape Analysis*. J. Wiley, 1998. ISBN: 9780471958161. URL: `http://books.google.de/books?id=71zvAAAAMAAJ`.

[ETDH07] Rainald M Ehrig, William R Taylor, Georg N Duda, and Markus O Heller. “A survey of formal methods for determining functional joint axes”. In: *Journal of Biomechanics* 40.10 (2007), pp. 2150–2157.

[Fre01] M. A. R. Freeman. “How the knee moves”. In: *Current Orthopaedics* 15.6 (2001), pp. 444–450.

[Goo91] Colin Goodall. “Procrustes Methods in the Statistical Analysis of Shape”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 53.2 (1991), pp. 285–339. URL: `http://www.jstor.org/stable/2345744`.

[HH95] Tony Heap and David Hogg. “Extending the Point Distribution Model using Polar Coordinates”. In: *In Proc. CAIP*. 1995, pp. 130–137. URL: `http://www.scs.leeds.ac.uk/dch/papers/heap/caip95.pdf`.

[Ipo] *Ipopt Homepage*. 2012. URL: `http://projects.coin-or.org/Ipopt`.

[KLZH09] Dagmar Kainmüller, Hans Lamecker, Stefan Zachow, and Hans-Christian Hege. “An Articulated Statistical Shape Model for Accurate Hip Joint Segmentation”. In: *EBMC 2009. Int. Conf. of the IEEE Eng. in Med. and Biol. Society (EMBC)*. Minneapolis, USA, Sept. 2009, pp. 6345–6351.

[KWLFO08] Tobias Klinder, Robin Wolz, Cristian Lorenz, Astrid Franz, and Jörn Ostermann. “Spine segmentation using articulated shape models”. In: *MICCAI* (2008), pp. 227–234. URL: `http://www.springerprofessional.de/028---spine-segmentation-using-articulated-shape-models/1446576.html`.

[LSHD04] Hans Lamecker, Martin Seebaß, Hans-Christian Hege, and Peter Deuflhard. “A 3D Statistical Shape Model of the Pelvic Bone for Segmentation”. In: *Proceedings of SPIE - Medical Imaging: Image Processing*. Vol. 5370. 2004, pp. 1341–1351. URL: `http://www.zib.de/lamecker/publications/spie2004.pdf`.

[Min00] Thomas P. Minka. *Old and New Matrix Algebra Useful for Statistics*. Tech. rep. Dec. 2000. URL: `http://research.microsoft.com/en-us/um/people/minka/papers/matrix/`.

[Sha05] A.A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, 2005. ISBN: 9780521850117. URL: `http://books.google.de/books?id=Bjye4r1j9ZIC`.

[Shl05] Jonathon Shlens. “A Tutorial on Principal Component Analysis”. In: *Systems Neurobiology Laboratory, Salk Institute for Biological Studies*. 2005. URL: `http://www.snl.salk.edu/~shlens/pca.pdf`.

[SKHLZH08] Heiko Seim, Dagmar Kainmüller, Markus Heller, Hans Lamecker, Stefan Zachow, and Hans-Christian Hege. “Automatic Segmentation of the Pelvic Bones from CT Data Based on a Statistical Shape Model”. In: *Eurographics Workshop on Visual Computing for Biomedicine (VCBM)*. Delft, Netherlands, 2008, pp. 93–100. URL: `http://www.zib.de/ramm/publications/vcbm2008.pdf`.

[WB06] Andreas Wächter and Lorenz T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106.1 (2006), pp. 25–57. URL: `http://dx.doi.org/10.1007/s10107-004-0559-y`.

Statement of authorship

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this nor a similar work has been presented to an examination committee.

Berlin, April 2, 2013 . . . .