



HAL
open science

Virtual restoration of digitized statues

Tong Fu

► **To cite this version:**

Tong Fu. Virtual restoration of digitized statues. Modeling and Simulation. Université de Lyon, 2021. English. NNT : 2021LYSE1123 . tel-03828680

HAL Id: tel-03828680

<https://theses.hal.science/tel-03828680v1>

Submitted on 25 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Université Claude Bernard



Lyon 1

N° d'ordre NNT : 2021LYSE1123

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de :

l'Université Claude Bernard Lyon 1

Ecole Doctorale ED512

Informatique et Mathématique de Lyon

Spécialité de doctorat : Informatique

Soutenue publiquement le 6 juillet 2021, par :

Tong Fu

Restauration virtuelle de statues numérisées

Virtual restoration of digitized statues

Devant le jury composé de :

Stefanie HAHMANN

Professeure, Grenoble INP

Mathias PAULIN

Professeur, Université Paul Sabatier

Franck HÉTROUY-WHEELER

Professeur, Université de Strasbourg

Yoann LAFON-JALBY

Maître de Conférences, Université Claude Bernard Lyon 1

Emmanuelle ROSSO

Maître de Conférences, Université Paris-Sorbonne

Raphaëlle CHAINE

Professeure, Université Claude Bernard Lyon 1

Julie DIGNE

Chargé de Recherche, CNRS

Rapporteure

Rapporteur

Examineur

Examineur

Examinatrice

Directrice de thèse

Co-Directrice de thèse

Université Claude Bernard - LYON 1

Président de l'Université	M. Frédéric FLEURY
Président du Conseil Académique	M. Hamda BEN HADID
Vice-Président du Conseil d'Administration	M. Didier REVEL
Vice-Président du Conseil des Etudes et de la Vie Universitaire	M. Philippe CHEVALLIER
Vice-Président de la Commission de Recherche	M. Jean-François MORNEX
Directeur Général des Services	M. Pierre ROLLAND

COMPOSANTES SANTE

Département de Formation et Centre de Recherche en Biologie Humaine	Directrice : Mme Anne-Marie SCHOTT
Faculté d'Odontologie	Doyenne : Mme Dominique SEUX
Faculté de Médecine et Maïeutique Lyon Sud - Charles Mérieux	Doyenne : Mme Carole BURILLON
Faculté de Médecine Lyon-Est	Doyen : M. Gilles RODE
Institut des Sciences et Techniques de la Réadaptation (ISTR)	Directeur : M. Xavier PERROT
Institut des Sciences Pharmaceutiques et Biologiques(ISBP)	Directrice : Mme Christine VINCIGUERRA

COMPOSANTES & DEPARTEMENTS DE SCIENCES & TECHNOLOGIE

Département Génie Electrique et des Procédés (GEP)	Directrice : Mme Rosaria FERRIGNO
Département Informatique	Directeur : M. Behzad SHARIAT
Département Mécanique	Directeur : M. Marc BUFFAT
Ecole Supérieure de Chimie, Physique, Electronique (CPE Lyon)	Directeur : Gérard PIGNAULT
Institut de Science Financière et d'Assurances (ISFA)	Directeur : M. Nicolas LEBOISNE
Institut National du Professorat et de l'Education	Administrateur Provisoire : M. Pierre CHAREYRON
Institut Universitaire de Technologie de Lyon 1	Directeur : M. Christophe VITON
Observatoire de Lyon	Directrice : Mme Isabelle DANIEL
Polytechnique Lyon	Directeur : Emmanuel PERRIN
UFR Biosciences	Administratrice provisoire: Mme Kathrin GIESELER
UFR des Sciences et Techniques des Activités Physiques et Sportives (STAPS)	Directeur : M. Yannick VANPOULLE
UFR Faculté des Sciences	Directeur : M. Bruno ANDRIOLETTI

RÉSUMÉ EN FRANÇAIS

La numérisation des objets archéologiques est une tâche importante pour la préservation et la transmission du patrimoine culturel, mais il est également indispensable de s'occuper du problème de leur restauration. Cependant, les processus de restauration physique traditionnels sont difficiles, coûteux et parfois risqués pour l'intégrité des vestiges eux-mêmes, en raison de mauvais choix de restauration par exemple. Pour éviter cela, il est intéressant de se tourner vers la restauration virtuelle, qui permet de tester des hypothèses de restauration, qui pourront être réalisées ultérieurement sur les vestiges réels. Dans ce contexte, notre objectif est de développer une nouvelle approche de restauration virtuelle de statues numérisées, fondée sur l'utilisation d'un modèle unifié gardant le lien avec la précision originale des scanners utilisés et permettant d'échafauder différentes hypothèses.

Cette thèse présente un cadre pour compléter les parties manquantes de statues archéologiques représentant des personnages ou des animaux.

Tout d'abord, nous proposons l'algorithme FAKIR qui permet de recalcr un modèle anatomique articulé de type sphere-mesh sur un nuage de points résultant de la numérisation d'une statue, en découvrant conjointement les paramètres anatomiques utiles à l'étalonnage du modèle. Il s'agit en effet d'un problème d'optimisation pour lequel il est possible d'exploiter des hypothèses de recalage rigide par morceaux, tout en s'inspirant d'algorithmes de cinématique inverse. Nous avons également testé les capacités actuelles offertes par l'apprentissage profond pour le recalage de squelette 2D et 3D afin de comparer et de mieux positionner l'algorithme FAKIR. La comparaison étaye notre intuition que l'apprentissage effectué à partir des images ou des relevés de vrais modèles humains ne permet pas de détecter avec justesse le squelette d'une statue. De plus, l'approche de recalage FAKIR se généralise à des modèles non humanoïdes, pour lesquels on ne dispose généralement pas de base d'apprentissage.

Ensuite, nous proposons deux nouvelles approches de skinning pour modifier la position d'une statue décrite par un nuage de points après que l'on ait identifié sa structure anatomique en utilisant FAKIR. La première approche adoptée distingue les mouvements de torsion, des mouvements de dépliement ou repliement d'une articulation entre deux membres, en prenant en compte l'anisotropie propre à ces articulations, mais reste fondée sur l'utilisation de poids pour déterminer le déplacement d'un point influencé par plusieurs os. La seconde approche de skinning utilise une description du détail de la statue au dessus d'un ensemble de lignes couvrant notre modèle articulé, avec une approche géométrique pour déterminer l'évolution des lignes et du détail suite à une déformation.

Enfin, sur la base de ces contributions, nous proposons une méthode pour combiner entre elles des parties de différentes statues après les avoir ramenées à une pose et une morphologie communes. Étant donné une statue à restaurer, nous commençons par identifier les parties manquantes. Des statues compatibles sont ensuite fournies par les utilisateurs afin de pouvoir compléter la statue à restaurer par combinaison la plus harmonieuse possible. Pour cela, la morphologie et la pose des statues sont modifiées au préalable pour correspondre à celles de la statue cassée. Une des originalités des approches développées est qu'elles opèrent directement sur des nuages de point, sans recourir à un maillage, afin de préserver la richesse de la précision fournie par les scanners.

Mots clés: restauration virtuelle, nuage de points, recalage, skinning sur nuage de point, déformation de forme, modèle de sphère-mesh,

ABSTRACT

The digitization of archaeological objects is an important task for the preservation and transmission of cultural heritage, but it is also essential to deal with the problem of their restoration. However, traditional physical restoration processes are complicated, costly, and sometimes risky for the artifact itself due to poor restoration choices. It is interesting to turn to virtual restoration, which allows testing restoration hypotheses, can later carry that out on the real artifact. In this context, our objective is to develop a new approach to the virtual restoration of digitized statues, based on a unified model keeping the link with the original precision of scanners and allowing to build different hypotheses.

This thesis presents a framework to complete the missing parts of archaeological statues representing characters or animals.

First of all, we propose the FAKIR algorithm which makes it possible to register an articulated anatomical model of the sphere-mesh type on a point cloud resulting from the digitization of a statue, by jointly discovering the anatomical parameters useful for the calibration of the model. This is indeed an optimization problem for which it is possible to exploit piecewise rigid registration assumptions while being inspired by inverse kinematics algorithms. We also tested the current capabilities offered by deep learning for 2D and 3D skeletal registration to compare and better position the FAKIR algorithm. The comparison supports our intuition that learning from images or scans of real human models does not accurately detect a statue's skeleton. Besides, FAKIR registration approach is generalized to non-humanoid models, for which there is generally no basis for learning.

Second, we propose two new skinning approaches to modify the position of a statue described by a point cloud after its anatomical structure has been identified using FAKIR. The first approach adopted distinguishes torsion movements, unfolding or folding movements of a joint between two limbs, by taking into account the anisotropy specific to these joints, but remains based on the use of weights to determine the displacement of a point influenced by several bones. The second skinning approach uses a description of a statue's detail above a set of lines covering our articulated model, with a geometric approach to determine the evolution of lines and detail following deformation.

Finally, based on these contributions, we propose combining different statues after bringing them to a common break and morphology. Given a statue to be restored, we start by identifying the missing parts. Compatible statues are then provided by the users in order to be able to complete the statue to be restored by the most harmonious combination possible. For this, the

morphology and the pose of the statues are modified beforehand to correspond to those of the broken statue.

One of the originalities of our approaches is that they operate directly on point clouds, without resorting to a mesh, in order to preserve the wealth of precision provided by the scanners.

Key words: virtual restoration, point cloud, registration, point set skinning, shape deformation, sphere-mesh model

ACKNOWLEDGEMENT

J'aimerais tout d'abord remercier mes encadrantes, Raphaëlle Chaine et Julie Digne, pour m'avoir guidé dans le monde scientifique, pour tous les temps qu'elles ont sacrifiées tout au long de ce travail de recherche, pour leur gentillesse et intelligences.

Je remercie également mes amis, mes collègues: Thomas, Laura, Léon, Sarah, qui étaient dans le même bureau avec moi, qui ont décorés mes jours monotones.

J'adresse aussi mes remerciements à l'équipe administrative pour m'avoir aidé sur mes dossiers.

Merci à Grégoire pour sa tolérance à mes mauvaises humeurs et pour son accompagnement pendant ces années. Merci à Miya pour sa méchanceté et sa gentillesse.

TABLE OF CONTENTS

Introduction	11
1 State of the Art	17
1.1 Anatomical Model	18
1.2 Skeleton registration	20
1.3 surface skinning	22
1.4 Pose change and shape synthesis	24
2 FAKIR: Anatomical model registration	25
2.1 Human model	25
2.2 Distance between the model and a point set	27
2.2.1 Distance from a point to one bone.	28
2.2.2 Distance from a point set to a sphere-mesh chain.	30
2.3 Optimization	31
2.3.1 Optimization for one bone	31
2.3.2 Optimization for a joint between two consecutive bones	33
2.4 FAKIR : Forward And bacKward Iterative Registration	35
2.4.1 Registration process for a chain of bones.	35
2.4.2 Full Skeleton Registration	38
2.5 Results	39
2.5.1 Experiments on synthetic data	39
2.5.2 Skeleton registration results on statues	40
2.5.3 Computation time	43
2.6 Discussion	43
2.6.1 Limitations	43
2.6.2 Importance of the optimization order for registering a chain of bones.	44
2.6.3 Importance of the normal-constrained projection	44
2.6.4 Automatic initialization test	45
2.7 Conclusion and perspectives	45
3 A first approach to sphere-mesh model skinning	55
3.1 Heightfield skinning	56

TABLE OF CONTENTS

3.1.1	Heightfield over the sphere-mesh model	56
3.1.2	Continuity	56
3.2	Base-point skinning	57
3.2.1	Bending rotation with anisotropic weights	59
3.2.2	Twisting rotation	61
3.2.3	Combination of twist and bend.	61
3.3	Results	62
4	A second approach to sphere-mesh model skinning	65
4.1	Baseline	66
4.1.1	Baseline before transformation	66
4.1.2	Baseline after transformation	67
4.2	Base-point	69
4.2.1	Base-point after transformation	69
	Interpolation for base-point position	70
	Interpolation for detail direction of base-point	70
4.3	Baseline skinning process	71
4.4	Geometry	72
4.4.1	The intersection of two cones	72
4.4.2	The cutting plane after transformation	73
4.5	Results	74
4.6	Conclusion and perspectives	76
5	Virtual Restoration results	79
6	Conclusion	85
	Bibliography	87
	List of publications	95
	List of figures	102

INTRODUCTION

Context

Today we are quite aware of the disappearance of world heritage. It is imperative to bear witness to these works by conserving, restoring, and showing them. With the progress of 3d scanning techniques, it is now common to create digital replicas of artworks, which will remain forever intact. Simultaneously, their real-world counterparts will slowly decay due to time damage or human activity. 3D acquisition of archaeological artifacts has become an essential part of cultural heritage research for preservation or restoration purposes. Statues, in particular, have been at the center of many projects. However, when the digitization is performed, the sculptures are often already degraded and need to be restored.

The e-ROMA project is on the theme of cultural heritage, aims to digitize, restore and animate statues and artworks, take advantage of the context of sites where they have occupied and the times they have passed. The problem of restoring statues is significant for the Gallo-Roman museum of Lyon-Fourvière, which is a partner in the project. Indeed, the museum has an important but very fragmentary collection of stone reliefs, testifying the high degree of Lugdunum Romanization. The museum now wishes to take advantage of the digital revolution to reconstruct a number of them and imagine the re-population of its architectural elements into statues that have now disappeared. This thesis is a part of the e-ROMA project, which concentrates on the virtual restoration of digital statues while keeping the original precision of data.

The first axis of this thesis consists of revisiting the problem of digitization and 3D modeling of artworks by putting precision, plausibility, and interactivity at the center of concerns and by associating it to real possibilities of virtual restoration, respecting the style of the time, with traceability and control of modifications done. The first step is to allow better use of laser surveys without restricting it only to 3D scanning and reconstruction steps. It is then a question of making full use of existing works and complementary knowledge that may relate to human anatomy or the behavior of the drape of fabrics, or even the history of canons in the art field, gestures, and sculpture tools of the time.

The second axis of the thesis focuses on generating new statues corresponding to new poses, physiognomies, or even to other trades or social positions to complete and give meaning to various architectural elements from excavations. Therefore, the notion of variety is essential and the ease with which a restorer or sculptor can obtain a new virtual model while remaining within

the framework of plausible hypotheses.

Thesis summary

Restoration requires a comprehensive understanding of archaeology and art history. Many recommendations have been made for guiding the restoration process [ICO64]. Manual restorations being costly, invasive and sometimes even risky, museums are often reluctant to carry out such processes. On the contrary, a virtual restoration process gives restorers the possibility to build and test different hypotheses with minimum user intervention.

While research has mainly focused on reassembling fractured objects [ED17, HFG*06], this thesis focuses on the virtual restoration of incomplete human statues, provided as 3D point sets. Nowadays, most of the virtual heritage restoration actions have been carried out by putting together the results obtained with decorrelated software suites. Sometimes, partial results are low, and tasks are repetitive and slow. The proposed framework in this thesis finds its roots in the work of Grossman et al. [GPT03] who discuss varying approaches to the restoration of ancient sculptures. In particular, they discuss restoration through the combination of ancient fragments, a technique widely used in the 18th and 19th centuries. Our virtual restoration follows this idea by harmoniously combining parts belonging to different statues after bringing them to a common pose and anatomy. The process contains three stages which we introduced in the following chapters.

- Stage 1: Pose and anatomy estimation of statues.
- Stage 2: Pose and anatomy changes.
- Stage 3: Completion by a combination of different statues.

First, we apply the FAKIR [FCD20b] method to register an articulated model to the statue point set. Second, each input point is represented by its residual displacement above the registered anatomical model. This point set encoding is coupled with a skinning method to modify statue poses and proportions. All steps are achieved directly on the point cloud, avoiding thus tedious meshing steps and preserving initial sampling accuracy. This virtual restoration has two advantages. First, there is no limit to the number of restoration hypotheses test. Second, completion can be achieved using different morphology and poses statues, avoiding thus adding an oversized arm to a little body. This thesis achieved a new approach of virtual restoration of digitized statues, based on the use of a unified model that keeps the link with the original precision of scanners and allows to build different hypotheses, to simulate evolutions over time and also to be interactive.

The following summarizes the content of each chapter:

Chapter 1: State of the art

This chapter develops state of the art on related works of virtual statue restoration. The three-stage framework proposed in this thesis reviews research on human models, skeleton registration, skinning algorithm, inverse kinematic, and shape synthesis.

Chapter 2: FAKIR: Anatomical model registration

This chapter introduces a way to improve the understanding of acquired statues representing real or imaginary creatures by registering a simple and pliable articulated model to the raw point set data. Our approach performs a Forward And bacKward Iterative Registration (FAKIR) which proceeds joint by joint, needing only a few iterations to converge. The model we use is inspired by the sphere-mesh model [TGB13]. The registration process iterates alternatively forward and backwards on kinematic chains. In spirit, this is similar to inverse kinematics, such as Fabrik [AL11] and CCD [WC91] algorithms, which define kinematic chains and compute their transformation from an input pose to a target pose by updating pose parameters one after the other alternatively forward and backward along each chain. However, we can detect the pose and the elementary anatomy of sculptures, possibly non-realistic body proportions. Our method can work on animals and imaginary creatures (Figure 1).

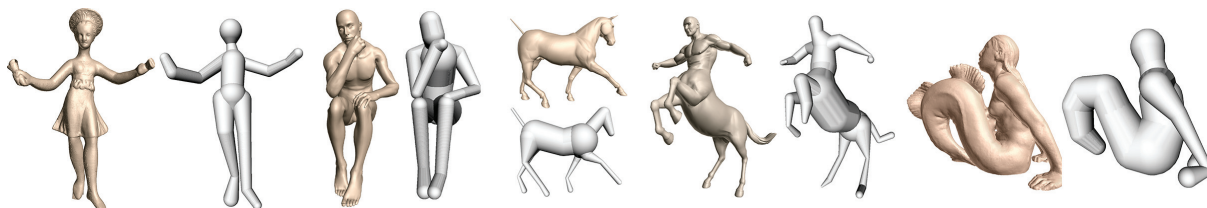


Figure 1 – FAKIR is able to detect the morphology and pose of articulated shapes, given an elementary anatomical model, from a single static scan. It is equally effective for human, animal shapes and even imaginary creatures.

We use the FAKIR algorithm to estimate the pose and anatomy of statues in the first stage of our virtual restoration.

The contributions are the following:

- A simple articulated model efficiently representing a statue pose and anatomy.
- An efficient calibration and registration process based on inverse kinematics principles.

Chapter 3: A first approach to sphere-mesh model skinning

We present our first approach of point set skinning using our sphere-mesh model in this chapter. While most skinning techniques work on a mesh, our skinning works directly on a point set for keeping the original precision of scan. Each input point is represented by its residual

displacement above the registered anatomical model. This point set encoding is coupled with an alternative to classical skinning method [BP07, TGB13]. The introduced skinning approach considers a movement as a combination of bending and twisting rotation. We compute bones' weights automatically. Weights are different for bending and twisting. We can use the presented method to modify statue poses and proportions in the second stage of our virtual restoration framework.

The contributions are the following:

- An approach to encode the point set to sphere-mesh model by heightfield.
- A point set skinning process with bending and twisting movement.

Chapter 4: A second approach to sphere-mesh model skinning

We introduce another skinning approach to the point set in this chapter. We provide a skinning technique that gives a realistic result and keeps the original precision for a statue's point set. This approach combines a point set to a registered sphere-mesh model so that deform the model will deform the point set accordingly. We propose associating a base-point on the articulated model to each of the original points. A baseline drives each base-point on the model and the position of a base-point after deformation follows its baseline movement. The new position of an input point is then deduced from its corresponding base points new position by reporting their initial detail values. Our approach's originality is to avoid using a mesh with fixed connectivity whose triangles quality may be altered by deformations related to pose and anatomy changes, possibly creating triangle slivers and self-intersections. Furthermore, we do not need to give weights of influenced bones for each point.

Chapter 5: Virtual restoration results

This chapter gives the full process for virtual restoration. We call a damaged statue to restore as the target statue. Statues who are chosen to complete the target statue are candidate statues. We introduce the method to complete missing parts of the target statue by combining the candidate statues' corresponding parts. Figure 2 shows an entire restoration process on the statue Prince Paris. We also show other restoration results of some impressive incomplete statues in this chapter.

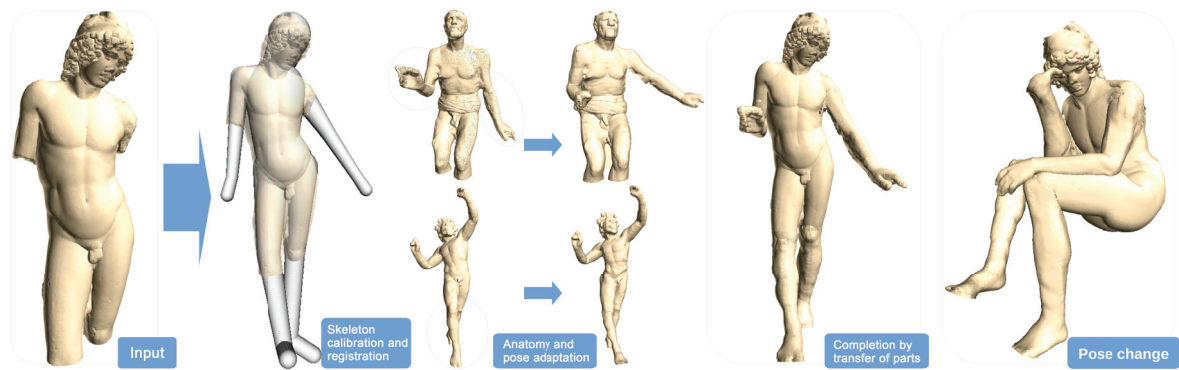


Figure 2 – Restoration and pose change of the Prince Paris statue. Through articulated model regression, we identify the anatomy and pose. We then adapt parts from other statues after a change of morphology and use them to complete the input. Once the statue is complete, we can also change its pose: here we bring it to the pose of *The Thinker* by Auguste Rodin.

STATE OF THE ART

The advent of laser scanners marked the digitization campaign's start, making it possible to obtain digital models of specific artworks and buildings. The digitization has many advantages for libraries and museums who wish to implement it, compared to the conservation or dissemination techniques previously used (such as photography, micro-forms): quality of the restitution, transfer and safeguard of data, possibility of reworking digitized documents to facilitate their appropriation by the public and the work of researchers. Digitization provides a new means of access to heritage, both for the public and for researchers, thanks to the possibility of networking digitized data. The digitization is useful when the shape has been well preserved. Otherwise, it is better to rely on art historians and restorers. In the formation of digital models, it is necessary to go much further than the cultural heritage's simple digitization. We need to move towards real virtual restoration actions. The application of geometry processing for virtual restoration in the cultural heritage has exciting potential for future research. Some virtual restoration has been achieved physically by digital fabrication support thanks to the 3d printing technology, such as the completion of vessels or vases [AEa*11], the restoration of a damaged medieval skull [FDCP*08] (See in Figure 1.1). Researches have focused on reassembling fractured objects [ED17, HFG*06] or fragmented statue [ASC*13] (See in Figure 1.2). However, these restorations are based on reassembling the existed pieces. They are not suitable for restoring statues whose missing parts are no longer existed.

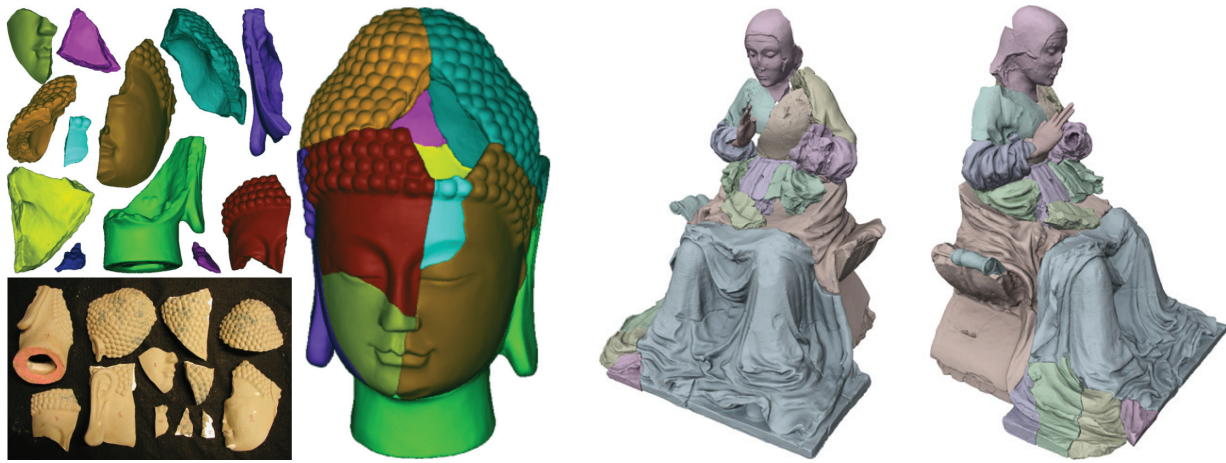
This thesis proposes a virtual restoration framework for incomplete human statues that the missing parts can not be found. The proposed framework in this thesis finds its roots in the



(a) Restoration of a bowl [AEa*11]

(b) Restoration of a medieval skull [FDCP*08]

Figure 1.1 – Restoration by 3d printing technology



(a) Reassembling a fractured head model [HFG*06] (b) Virtual assembly of the fragments of a statue [ASC*13]

Figure 1.2 – Virtual restoration by reassembling fractured pieces

work of Grossman et al. [GPT03] who discuss varying approaches to the manual restoration of ancient sculptures. The restorer’s widely used technique is to combine parts belonging to different statues harmoniously. For example, the Diana statue’s (Figure 1.3) head and torso date from antiquity and come from different statues, all other parts were manufactured in the 19th century.

1.1 Anatomical Model.

Designing anatomical models for human shapes has raised much interest in Computer Graphics. The most common representation consists in a more or less detailed graph of bones such as the ones used in the MakeHuman framework [Bas00], while some methods go beyond this kind of elementary skeleton representation and model every single muscle to increase realism [LGK*12]. We will focus on basic skeletons, which are pliable and efficient enough for our purpose due to our nonrealistic context. A skeleton-based model consists of two components: a skeletal structure [BK00] and a representation for the volume surrounding it. This representation can be either mesh surfaces or volumetric primitives. Recent surface-based models [ASK*05, PMRMB15, LMR*15, HSS*09, ZB15] are learned from numerous scans of real people. We show examples in Figure 1.4. Among those, SMPL [LMR*15] is a human anatomical mesh model in which a set of parameters control non-rigid deformations resulting from a statistical study on a large number of humans and positions. However, to our knowledge, no approach allows us to position this model from a static point cloud without positioning the model close to the data or without using 2D views and deep learning. It works well to cap-



Figure 1.3 – Diana, goddess of the hunt. It is restored in 2013.

ture human motion and shapes in a video [ZPBPM17] or images [BKL*16, HBL*17]. But this model has a poor performance when the data is unrealistic which is common for archaeological statues. Another possible shape representation is based on volumetric primitives, e.g. using medial axis transform (MAT) [Blu67, SCYW15], metaballs [PF01], B-meshes [JLW10] or other primitives [GD96, SBR*04, ARM*19] (See in Figure 1.5). Among these models, the sphere-mesh model [TGB13], a variant of convolution surfaces [BS91], has been introduced for representing mesh models by packing spheres into it and encoding their structure. Conceptually, the sphere-mesh model can be seen as a piecewise linear simplification of the computational geometry skeleton [TDS*16]. Although sphere-meshes were developed to extract the shape structure from an input mesh, they can represent an anatomical model by imposing constraints. It has been used successfully for representing hand skeletons [TPT16, RTTP17]. This model is light and pliable, and we will also rely on it.

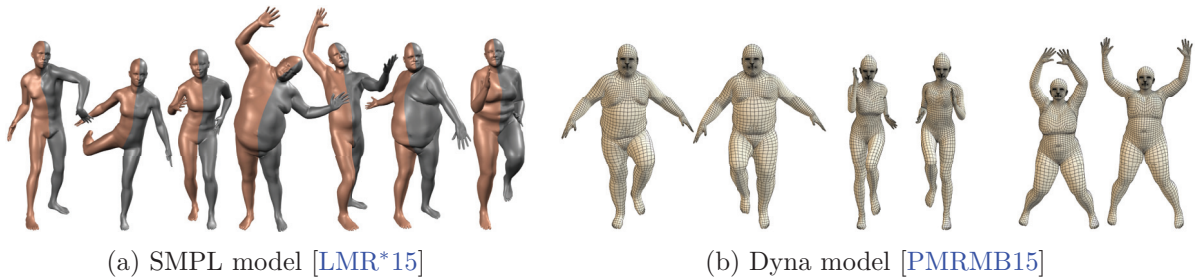


Figure 1.4 – Two human models in different shapes and poses

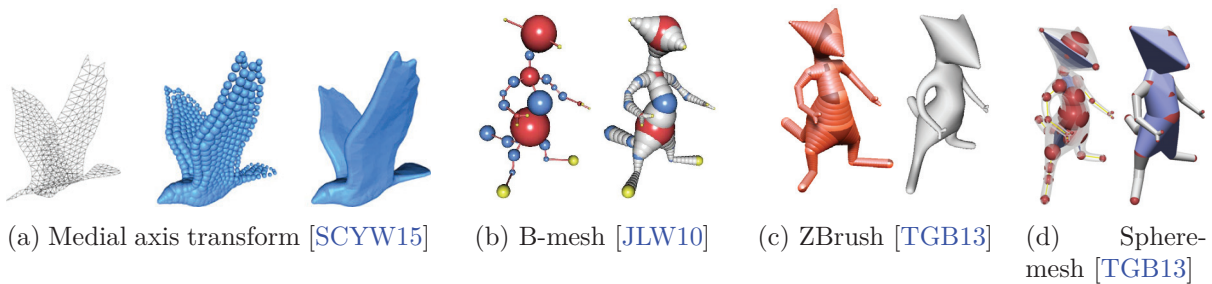


Figure 1.5 – Examples of shape representation based on volumetric primitives

1.2 Skeleton registration

Registering a model to a shape is an important task which has received much research interest. The goal can be to animate a shape by skeleton *rigging* and *skinning* or to detect human poses. Skeleton rigging can be performed manually [BK00, JLW10], but a few methods have investigated automatic processes. In particular, the Pinocchio algorithm [BP07] adapts a skeleton to a static mesh by defining an objective function and maximizing it (Figure 1.7a). It works by packing spheres into the mesh and considering their centers, gathered in a graph, as the admissible joint positions. This pre-computation makes the skeleton pose estimation tractable. On points sets, a ℓ^1 -medial skeleton could be used [HWCO*13] alternatively to sphere-packing (Figure 1.7b), but it is not suitable for noisy or incomplete data, as shown by our experiments (Figure 2.18).

If the input data is dynamic, it is possible to infer or track, a skeleton. Most tracking approaches [SBB10, GSDA*09, TZMS04] focus on the capture of the positions of the joints and deduce pose parameters (angles) and intrinsic parameters (*e.g.* bone lengths) from it. Many of such tracking methods [SHG*11, WZC12] start with a calibrated skeleton, but the calibration itself can be performed from a depth video and a set of known admissible poses [TPT16, RTTP17, TTR*17]. Such methods require a dynamic scene and cannot apply to the static shape

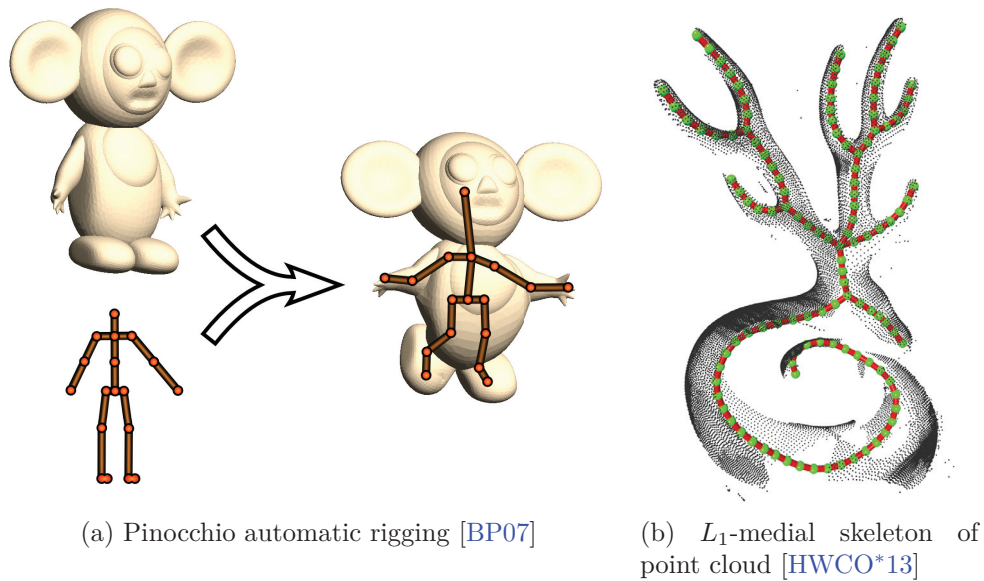


Figure 1.6 – Different methods to discover the skeleton.

rigging problem. It is also possible to rely on a database of people scans to learn the pose and deformation of human bodies using the SMPL model [HLRB12, WHB11, AMX*18]. Recently, CNN-based detectors, such as DeepCut[PIT*16] and OpenPose[CSWS17] were used for 2D joints detection in images or videos. Bogo et al. [BKL*16] estimate the 3d human pose and proportions from a single image by fitting an SMPL model to DeepCut estimated joint positions. Using multi-view images over time [HBL*17] improves the pose accuracy. However, human proportions remain approximate. Human tracking can also be done without needing a model [BBLR15]. Similarly, it is possible to register two models using manifold-harmonics based non-rigid registration [LRB*16], but this would not help for skeleton-based registration. Learning approaches can also work from a single image [LIPM19, AMB*19].

Registering an SMPL skeleton directly to a point cloud has been addressed using deep learning directly on point sets augmented with feature detection [JCZ19], but this method only targets human shapes, which it learns from a database. In contrast, our method can work on nonrealistic anatomies and various animals, as demonstrated in our experiments. Deforming a point cloud to match a template mesh has also been tackled using auto-encoders [LSS*19], but this requires a full template mesh for each model. Our required skeleton model is much lighter. Finally, recently, the FARM[MMRC20] method builds on the functional map framework to register a parametric model (such as the SMPL one) to a mesh or a point set in a fully automatic way. This approach reaches the state of the art results while being the closest in goal to ours, and we will compare to it. Finally, some methods [HSR*09, ZPBPM17, PMPHB17, YZZ*19] aim at finding a person’s pose despite its sometimes loose clothing, but this is outside the scope

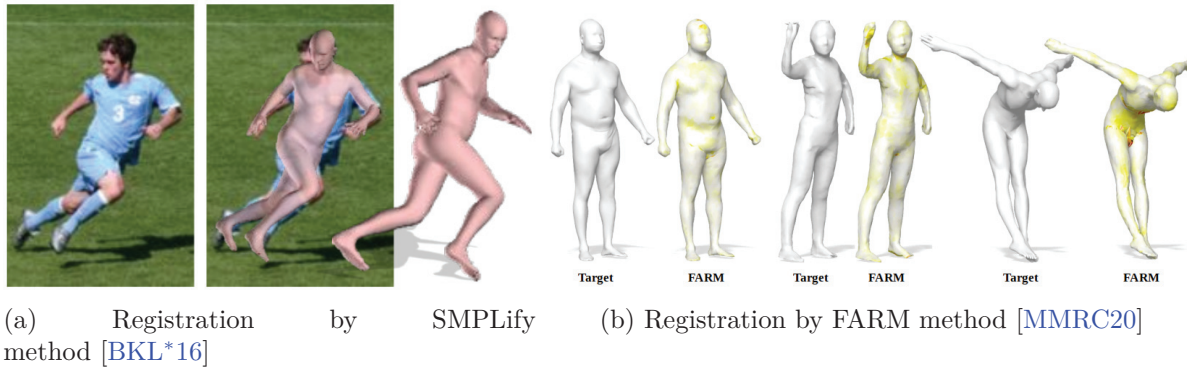


Figure 1.7 – Different registration methods.

of our paper [SBB10, GSDA*09, TZMS04].

Our registration algorithm makes extensive use of kinematic chains, processing them alternatively forward and backwards. In spirit, this is related to inverse kinematics, and in particular, the Fabrik [AL11] and CCD [WC91] algorithms. Indeed, both methods define kinematic chains and compute their transformation from an input pose to a target pose by updating pose parameters one after the other alternatively forward and backward along each chain. However, the similarity ends here, since our goal is to estimate the pose and the proportions of the model limbs using data-attachment constraints in a static framework.

1.3 surface skinning

Once the skeleton is correctly positioned, a skinning process can be used for pose change or animation purposes. The most famous and simple skinning method is called Linear Blend Skinning (LBS for short) [MtLTM88]. A vertex on a mesh surface point is transformed by a linearly weighted combination of the motions of the moving bones it is attached to. Despite well-known limitations (the candy wrapper effect and the elbow collapse effect), Linear Blend skinning is still the standard skinning method for real-time animation purposes. Many skinning approaches base on the principle of LBS and improve its limitations, such as Pose Space Deformation [LCF00], Log-matrix blending [Ale02, TCSP04], Optimized centers of rotation [LH16], Multi-Weight Enveloping [WP02], spherical Skinning [Kv05] and Dual Quaternions Skinning [KCZO08]. Kavan et al. [JS11, KSH12] decomposes the motion of a joint into bend and twist motions to reduce the artefacts of Linear Blend Skinning and Dual Quaternions Skinning. Our method [FCD20a] further improves the result by handling bend rotations anisotropically. But there are still some artefacts at joints such as a bulge effect.

Furthermore, setting the right weights is an important question for these skinning methods: while the profile of the weights is generally sketched by graphic designers [MTG03], there exist

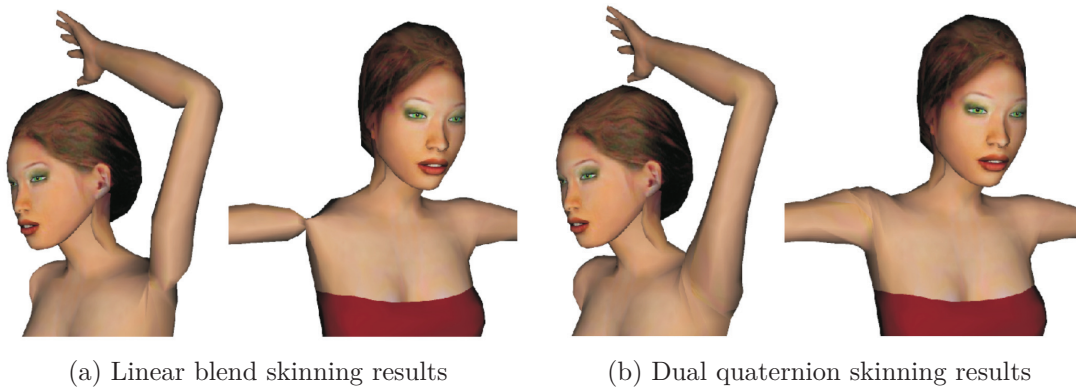


Figure 1.8 – Comparison of linear and dual quaternion blending. Images from [KCZO08]

automatic weighting techniques that, for example, use heat diffusion [BP07, TGB13, ARM*19], geodesic voxel binding [DL13] or set bounded biharmonic weights [JBPSH14]. Applying suitable weights on the transformation matrices of bones gives a smooth transition at bones joints when the joints are bending, but they are not sufficient to handle twisting or bone stretching. Our method doesn't require such weights computation and can deal with all these deformations in a unified way. A recent skinning method [LL19] corrects artefacts of Linear Blend Skinning by locally estimating the rigid transformation that best restores the relative position of a vertex with respect to its neighbors using Laplacian differential coordinates. This method, designed for meshes, involves a definition of details in terms of Laplacian differences. In our approach, we rather define the detail as the residual over the registered anatomical model. Instead of using a skeleton, some methods [JSW05, JMD*07] rely on a cage deformation which can deal with more general deformation like twisting or stretching. The deformations are controlled using a topologically flexible cage that consists of a closed three dimensional mesh. Cage deformations are designed for unrealistic articulate characters such as cartoon characters or objects and can work in 2D or 3D.

Going in a different direction, Physics-based methods simulate the growth of skeletal muscles and fat tissues using a biological model. Hamadi et al. [AHLG*13] transfers the volume delimited by a mesh to the interior of another mesh by minimizing some harmonic energy. To do so, a deformation field is computed between the two meshes, based on a nearest point matching which is updated at each iteration [GRP10]. However, the process is not fully automatic. Recent researches [SZK15, IKKP17, KIL*16] achieve a desired deformation of human body and face by direct control of each muscle. These methods are computationally expensive and need to define the physical model from physiological data. They are not suitable for artistic shapes since the morphology and the muscle of an artwork may be very different from a biological model.

Taking a different perspective on the problem, Implicit skinning [VBG*13] uses an implicit

formulation of the surface that better supports pose changes and re-projects skinned vertices on the implicit model after each pose change. In this paper, we also use a proxy model but it is explicit. Volume preserving skinning methods [FTS08, RHC09, AS07] correct for volume changes through the generation of extra bulges and wrinkles. They use vector fields induced by skeletal motion to describe the skin deformation.

Example-based methods [SRC01, WPP07, MG03, KJP02] produce more realistic results but require extra training data. They have a limitation for a given range of deformations. In addition, these methods can only be as good as their training dataset is, and for unrealistic characters (imaginary creatures or characters with unrealistic body proportions), relevant datasets are nontrivial to build.

1.4 Pose change and shape synthesis

When a model is rigged and skinned, it is possible to change its pose manually by interacting with the skeleton joints. Through the skinning weights, the mesh surface should deform accordingly. However, it is often tedious to design every single motion of each joint for each animation frame. Consequently, research has focused on inferring the movement from some critical joints and frames with given skeleton positions. In this inverse kinematics context, the Fabrik [AL11] and CCD [WC91] algorithms define kinematic chains and aim at transforming each chain from its input pose to its target pose by updating pose parameters one after the other alternatively forward and backward along the chain. Shape synthesis is often done by reusing existing models [FKS*04]. The idea is to retrieve suitable models from a database and warp the retrieved models to conform with the incomplete model [PMG*05, CK10]. A probabilistic representation for the components of a shape has been developed to suggest relevant components during an interactive assembly-based modelling session [CKGK11, KCKK12]. In our work, we propose a framework to statue shape synthesis by part combination, the choice of the parts being done manually while the part adaptation is automatic.

FAKIR: ANATOMICAL MODEL REGISTRATION

3D scanning techniques make it possible to create digital replicas of artworks. As part of the automatic processing of scanned statues, it is often necessary to identify the model’s pose and anatomy. Indeed, registering a model to a statue is useful for many applications. For example, one can bring statues to a common pose to better compare their style. It can also combine statue parts to restore broken statue virtually or animate a statue.

While pose recognition has been efficiently addressed for human models, in particular using Machine Learning algorithms, these methods can only work if the model fits the training dataset of shape models. A challenge of artistic human statues than real human scans lies in the difference in aesthetic perception. As we will see, this has drastic consequences for example-based machine learning algorithms, which fail at adapting to these statues. As for imaginary creatures, no such training database exist. Hence there is a need for simple anatomical models, that can be fitted without extensive training on various creatures, animals or humans with non-realistic body proportions.

We focus on human statues with no or few garments, animals, and imaginary creatures in this work. Furthermore, we consider that the digitized statues are provided as point sets. We propose a method for calibrating and registering a simple articulated model to a point set, named Forward And bacKward Iterative Registration (FAKIR). FAKIR works directly on the point cloud, avoiding the tedious meshing step. FAKIR iterates between assigning each point to its best corresponding model part, optimizing the anatomical model pose and proportions accordingly, and converges in only a few iterations.

2.1 Human model

Indeed many sculptors favoured the perceived beauty of their work over the realism of human proportions [Hus55, New34]. Figure 2.1 shows examples of such unrealistic statues of the Roman and Gallo-Roman eras. In this context, it is necessary to devise a human model with few constraints, allowing to fit a sculpture which does not follow the human proportion beauty canons. The existing fully detailed human templates modelling every single limb and muscle in a

very realistic way are too constrained for our purpose. In particular, we avoid modelling muscles in our model. After we have registered our simplified human model to the statue point cloud, a further step could be to add new flexible muscle models and is unnecessary for statues where the sculptor uses muscles as stylistic elements, in the same way, that he could use arabesques, yielding a possibly unrealistic result.



(a) A Gallo-Roman statue of a Gallic warrior (Avignon, France, picture: F. Philibert-Caillat)

(b) Statue of Heracles and Cacus (1530-1534) by Baccio Bandinelli, (Florence, Italy, picture: Cyberuly)

Figure 2.1 – Examples of statues with unrealistic anatomies.

We introduce an anatomical model inspired by the sphere-mesh model [TGB13], already successfully used for hand tracking [TPT16, RTTP17], using only one-dimensional elements. In this model, each bone is represented by a sphere-mesh B corresponding to the envelope of the union of a set of spheres centered on a segment and with a linearly varying radius (Figure 2.2b). Each bone is defined by two end sphere centers c_1 and c_2 with associated radii r_1 and r_2 respectively. The segment $[c_1c_2]$ is the medial axis of the bone. For each point $c \in [c_1c_2]$, the radius of the sphere centered at c is $r(c) = (1 - \tau)r_1 + \tau r_2$, with $\tau = \frac{\|c_1c\|}{\|c_1c_2\|}$.

The sphere-mesh model is controlled by the length $l = \|c_1c_2\|$ and the pair of sphere radii $\mathbf{r} = \{r_1, r_2\}$. Consequently, we denote the sphere-mesh model for one bone as $B(l, \mathbf{r})$. We also denote by α the angle of the conic part of the bone, as illustrated on Figure 2.2a. Significantly enough, the bones we are defining do not correspond to anatomical bones, but more to limbs (*i.e.* it includes a coarse description of the flesh volume around the anatomical bone). By analogy

to inverse kinematics, we keep the word *bone* instead of *limb*.

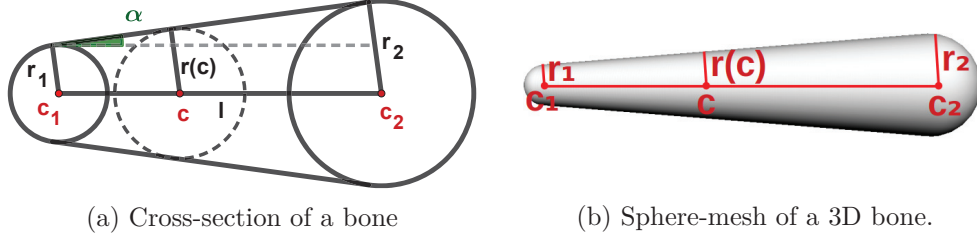


Figure 2.2 – The sphere-mesh of a bone is the union of the spheres centered on segment $[c_1c_2]$, with radius varying linearly between the two extremities of the segment.

With this type of bone element, we construct a simple human body template with very coarse respect of human proportions as an initial body shape (Figure 2.3a), but we can construct a template for any other animal or imaginary creature (Figure 2.3b) as well. Our human body template contains 22 bones $\{B_k\}_{k=1..22}$. Three bones correspond to the pelvis and have no relative motion: their length is fixed up to a common scale parameter that will be determined during the registration, along with the orientation of the triplet. Additionally, we use a particular bone to connect the spine bone to the neck, and its length and orientation directly depend on the adjacent spine bone. The other bones have no constraint on their relative proportions. The bones are organized into five chains, depicted in different colours in Figure 2.3: the spine chain, the right arm chain, the left arm chain, the right leg chain and the left leg chain. These chains are independent with the only constraint that some extremities must remain anchored to the spine. The chains' orientation is used to define the predecessor and successor of each bone. The ordering will be reversed to process the chain forward and backward several times during the registration process. Each bone is thus fully defined by its intrinsic parameters (length and two radii) and its extrinsic parameter (rotation with respect to its predecessor). Furthermore, two successive bones share a common radius. Because of the simplicity of the sphere-mesh bone model, the distance from a point to the model can be easily computed. In contrast, using a mesh model would make these computations much more demanding.

2.2 Distance between the model and a point set

To capture the anatomy and the pose of a statue, we need a distance function to measure how the sphere-mesh model fits a point set P , even if the points are far from their attached bones. The sphere-mesh model calibration and registration strive to reduce the distance between the sampled points and their corresponding bones. The problem is that the bone to which a point should be assigned is unknown, especially if the model has not been calibrated beforehand and if it is far from the data. Therefore, the target bone is usually replaced by the closest bone. We

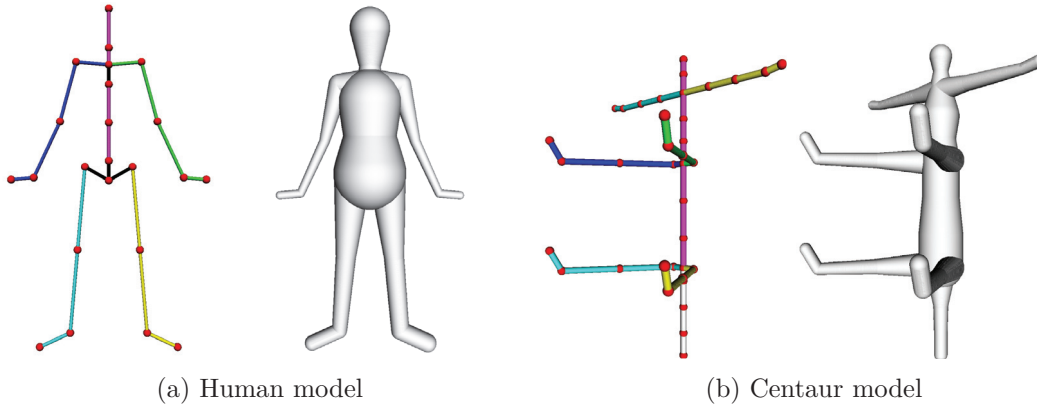


Figure 2.3 – Skeleton and sphere-mesh models for humans and centaurs. For the human model: the bones are organized into 5 chains shown in different colors. 4 additional bones are drawn in black: the pelvis which is a constrained triplet of bones, and the connection bone between the spine and the neck. Our model can also be adapted to various creatures, even imaginary ones such as a centaur.

assume that the points’ coordinates are provided with a coarse approximation of the oriented normal. This speeds up the registration when the model is not close to the data, but it remains possible to implement our registration algorithm with a simple Euclidean distance from the points to the model.

2.2.1 Distance from a point to one bone.

We start by defining the normal-constrained projection of a sampled point p on a single bone B by using the oriented normal vector \mathbf{n}_p to disambiguate the choice between several orthogonal projection possibilities. Given a point p in the ambient space with oriented normal \mathbf{n}_p , we consider the lines passing through p and orthogonally intersecting the sphere-mesh surface (possibly crossing its interior) at some points.

Whenever it is possible, we select the projection \tilde{p} whose normal $\mathbf{n}_{\tilde{p}}$ has positive scalar product with \mathbf{n}_p . Considering this normal-constrained orthogonal projection allows for a faster convergence and better results (see section 2.6.3 for more details). Since each point p has a normal-constrained projection on all the bones, we refer to its normal-constrained projection on bone B_k as \tilde{p}_k . If no subscript is provided, \tilde{p} refers to the normal-constrained projection of p on the closest bone.

We detail the projection of a point p on a bone B given an approximation of the oriented normal of point p . Instead of using the usual orthogonal projection on the bone, we constrain the projection \tilde{p} to have a normal coherent with the one of p . This constraint is helpful when the bone lies far away from its corresponding point set: the point can then be projected on the “right side” of the bone. In the following, without loss of generality, let us assume $r_1 \leq r_2$. All

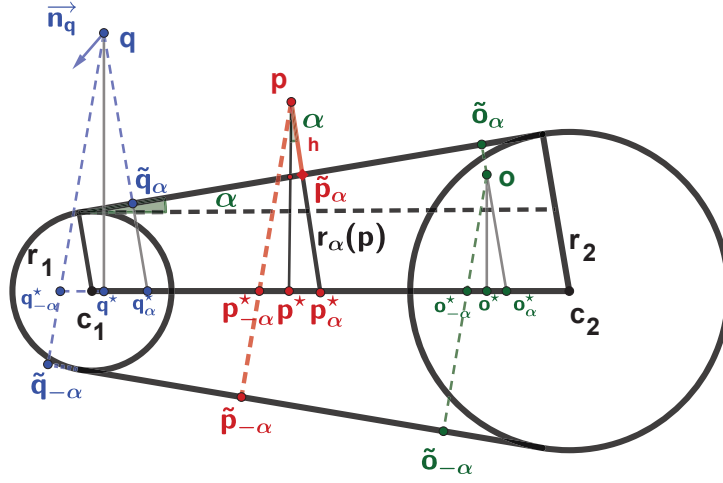


Figure 2.4 – Various projection cases. p has two possible projections \tilde{p}_α and $\tilde{p}_{-\alpha}$ depending on the orientation of the normal at p . Point p^* is the projection of p on line c_1c_2 . If the normal at p is oriented upward $\tilde{p} = \tilde{p}_\alpha$. Otherwise, $\tilde{p} = \tilde{p}_{-\alpha}$. The same strategy is used to project points q and o .

the following computations depend on an angle α defined in Fig. 2.4 and which can be expressed as $\alpha = \arctan \frac{|r_2 - r_1|}{\sqrt{\|c_1c_2\|^2 - (r_2 - r_1)^2}}$. Let us first compute p^* the projection of p on the oriented line c_1c_2 , and two translations of these points along this line: p_α^* at the distance $\|pp^*\| \tan \alpha$ of p^* and $p_{-\alpha}^*$ at the distance $-\|pp^*\| \tan \alpha$, as illustrated on Figure 2.4. Let $\tau_\alpha = \frac{p_\alpha^* - c_1}{c_2 - c_1}$, so that p_α^* can be expressed as $\tau_\alpha c_1 + (1 - \tau_\alpha)c_2$. Different cases can occur:

- $0 < \tau_\alpha < 1$: the point projects on the cone part of the bone. Let \tilde{p}_α be the intersection of segment $[p_\alpha^*p]$ with the cone. \tilde{p}_α is the orthogonal projection of p on the bone. If the normal to \tilde{p}_α has a positive scalar product with the normal of p , $\tilde{p} = \tilde{p}_\alpha$. Otherwise, normals are deemed inconsistent and $\tilde{p} = \tilde{p}_{-\alpha}$, *i.e.* the farthest intersection of $pp_{-\alpha}^*$ with the non-truncated cone. This situation occurs when the point p is on the wrong side of the bone (*i.e.* its normal is inconsistent with the normal of its closest point on the bone).
- $\tau_\alpha < 0$ (resp. $\tau_\alpha > 1$): \tilde{p} is the projection of p on the sphere centered at c_1 (resp. c_2) with consistent normal direction, except if this normal-constrained projection falls within the bone and not on the envelop. In that case, p is on the wrong side of the bone, and we set $\tilde{p} = \tilde{p}_{-\alpha}$ on the other side of the non-truncated cone.

In any case, the distance between p and its normal-constrained projection \tilde{p} vanishes when p is located near the surface of one bone, with a normal oriented consistently. It may happen that the returned projection does not provide a point belonging to the surface of the bone: on Figure 2.4, $\tilde{q}_{-\alpha}$ is the normal-constrained projection of point q , but it is not on the surface of the bone. It corresponds to a case where the point is very far from the part of the bone, which is coherent with its normal. During the registration process, $\tilde{q}_{-\alpha}$ will attract q on the other side

of the bone, so that a more consistent one will gradually replace the projection point.

For completeness, let us express unsigned distance $d(p) = \|p - \tilde{p}\|$ in the various cases since they will be required in the following Levenberg-Marquardt optimization formulations. If $\tau_\alpha < 0$ (resp. $\tau_\alpha > 1$), $d(p) = \|c_1 p\| - r_1$ (resp. $d(p) = \|c_2 p\| - r_2$). If $0 \leq \tau \leq 1$:

$$d(p) = \begin{cases} \|pp_\alpha^*\| - r_\alpha(p) & \text{if } n_{\tilde{p}} \cdot n_p > 0 \\ \|pp_{-\alpha}^*\| + r_{-\alpha}(p) & \text{if } n_{\tilde{p}} \cdot n_p \leq 0 \end{cases} \quad (2.1)$$

Since the radius of the cone varies linearly along line $c_1 c_2$:

$$\begin{aligned} r_\alpha(p) &= \|\tilde{p}_\alpha p_\alpha^*\| = (1 - \tau_\alpha(p))r_1 + \tau_\alpha(p)r_2 \\ r_{-\alpha}(p) &= \|\tilde{p}_{-\alpha} p_{-\alpha}^*\| = (1 - \tau_{-\alpha}(p))r_1 + \tau_{-\alpha}(p)r_2 \end{aligned} \quad (2.2)$$

with: $\tau_\alpha(p) = \frac{c_1 p_\alpha^* \cdot c_1 c_2}{\|c_1 c_2\|^2}$ and $\tau_{-\alpha}(p) = \frac{c_1 p_{-\alpha}^* \cdot c_1 c_2}{\|c_1 c_2\|^2}$. Furthermore $\|pp_\alpha^*\| = \|pp_{-\alpha}^*\| = \|pp^*\| / \cos \alpha$. Hence, for each bone, we first compute the α angle, then, for each point p , we compute its projection p^* on $c_1 c_2$ and the corresponding $\tau_\alpha(p)$ yielding $r_{\pm\alpha}(p)$ and $\tilde{p}_{\pm\alpha}$.

2.2.2 Distance from a point set to a sphere-mesh chain.

Given a point set P and a sphere-mesh chain of K bones, we first need to approximate the subset of current points that project on each bone. In the following, we define the point set P_k as the subset of points $p \in P$ which are closest to bone B_k using the distance $d_k = \|p - \tilde{p}_k\|$, $k = 1 \dots K$. Once the assignment is computed, the one-bone distance function E_k is defined as the sum of squared distances from points of P_k to bone B_k :

$$E_k(P_k, B_k(l_k, \mathbf{r}_k), \boldsymbol{\theta}_k) = \sum_{p \in P_k} \|p - \tilde{p}_k\|^2 \quad (2.3)$$

Importantly enough, the subset P_k and the one-bone energy E_k depend on the position of the initial extremity of the chain of bones involving B_k , as well as the parameters of the other bones in the chain.

The sum of one-bone distance functions measures the fitness of the model and serves as an objective function that we aim to minimize to capture the anatomy and pose of the sphere-mesh that best corresponds to our point set.

$$E = \sum_{k=1}^K \sum_{p \in P_k} \|p - \tilde{p}_k\|^2. \quad (2.4)$$

In the next sections, we will also be interested in the distance restricted to two adjacent

bones B_k and B_{k+1} , which we call two-bones energy:

$$E_{k,k+1} = \sum_{p \in P_k} \|p - \tilde{p}_k\|^2 + \sum_{p \in P_{k+1}} \|p - \tilde{p}_{k+1}\|^2. \quad (2.5)$$

2.3 Optimization

2.3.1 Optimization for one bone

The optimization of the one-bone energy is only used to estimate the parameters of the extremities of a chain, or for the first forward pass in Algorithm 1. After this step, the optimization of a joint's position using a bone pair should be preferred because it is more precise than the optimization of one bone.

In the single-bone case, the aim is to estimate the 3D rotation of the bone, its length and the radius of its free extremity by minimizing the one-bone energy $E_k(P_k, B_k(l_k, \mathbf{r}_k), \boldsymbol{\theta}_k)$, where $\boldsymbol{\theta}_k$ are the angles of rotation with respect to the predecessor's bone. This optimization is performed using the Levenberg-Marquardt algorithm for each parameter. In particular, the rotation can be decomposed into two rotations around two axes that are orthogonal to $c_k c_{k+1}$, indeed the rotation around $c_k c_{k+1}$ is not considered since it leaves the bone unchanged.

To optimize $\boldsymbol{\theta}_k$, we iteratively look for the best angle $\boldsymbol{\theta}_k + \delta\boldsymbol{\theta}_k$. At a minimum, $\nabla_{\delta\boldsymbol{\theta}_k} E_k(P_k, B_k(l_r, \mathbf{r}_k), \boldsymbol{\theta}_k + \delta\boldsymbol{\theta}_k) = \mathbf{0}$, and the value for $\delta\boldsymbol{\theta}_k$ follows. The details for the damped least-squares estimation are provided in the following.

Let us assume that c_1 (Fig. 2.4) is fixed and let us optimize for the pose and intrinsic parameters of bone B . In a local reference frame centered at c_1 with x-axis aligned with $c_1 c_2$, c_1 has coordinates $(0, 0, 0)$ and c_2 has initial coordinates $(l, 0, 0)$. The rotation of the bone can be parameterized by a rotation of angle θ_1 around the y-axis followed by a rotation of angle θ_2 around the z-axis. The one-bone energy is invariant by rotation around the x-axis. After the double rotation, c_2 has coordinates $(l \cos \theta_2 \cos \theta_1, l \sin \theta_2, l \cos \theta_2 \sin \theta_1)$. Let us call (x, y, z) the coordinates of point p in this local coordinate system and express $d(p)$ with respect to parameters $\boldsymbol{\theta} = (\theta_1, \theta_2), l$ and $\mathbf{r} = (r_1, r_2)$. We have:

$$\begin{aligned} \tan \alpha &= \frac{r_2 - r_1}{\sqrt{l^2 - (r_2 - r_1)^2}}, \cos \alpha = \frac{\sqrt{l^2 - (r_2 - r_1)^2}}{l} \\ \|c_1 p\|^2 &= x^2 + y^2 + z^2 \\ \|c_1 p^*\| &= x \cos \theta_2 \cos \theta_1 + y \sin \theta_2 + z \cos \theta_2 \sin \theta_1 \\ \|p^* p\|^2 &= x^2 + y^2 + z^2 - (x \cos \theta_2 \cos \theta_1 + y \sin \theta_2 + z \cos \theta_2 \sin \theta_1)^2 \\ \|p^* p_\alpha^*\| &= \|p^* p\| \tan \alpha \end{aligned}$$

$$\begin{aligned}\|p_\alpha^* p\| &= \frac{\|p^* p\|}{\cos \alpha} \\ \tau_{\pm\alpha}(p) &= \frac{\|c_1 p^*\| \pm \|p^* p_\alpha^*\|}{l} \\ \|c_2 p\|^2 &= (x - l \cos \theta_2 \cos \theta_1)^2 + (y - l \sin \theta_2)^2 + (z - l \cos \theta_2 \sin \theta_1)^2\end{aligned}$$

The one-bone energy function is (dropping the k subscript for simplicity):

$$E(P, B(l, \mathbf{r}), \boldsymbol{\theta}) = \sum_{\mathbf{p} \in \mathbf{P}} \mathbf{d}(\mathbf{p})^2 \quad (2.6)$$

The optimization is performed on three set of parameters in turn: angles $\boldsymbol{\theta}$, bone length l and bone radii \mathbf{r} .

the optimization for bone B with respect to $\boldsymbol{\theta}$ writes:

$$\hat{\boldsymbol{\theta}} \equiv \underset{\boldsymbol{\theta}}{\operatorname{argmin}} E(P, B(l, \mathbf{r}), \boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{p \in P} d(p, \boldsymbol{\theta})^2 \quad (2.7)$$

Following the Levenberg-Marquardt algorithm, at each iteration, parameter $\boldsymbol{\theta}$ is replaced by a new estimate $\boldsymbol{\theta} + \delta\boldsymbol{\theta}$, computed as:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} E(P, B(l, \mathbf{r}), \boldsymbol{\theta}) \approx \underset{\delta\boldsymbol{\theta}}{\operatorname{argmin}} E(P, B(l, \mathbf{r}), \boldsymbol{\theta} + \delta\boldsymbol{\theta}) \quad (2.8)$$

which is computed by taking:

$$\frac{\partial E(P, B(l, \mathbf{r}), \boldsymbol{\theta} + \delta\boldsymbol{\theta})}{\partial \delta\boldsymbol{\theta}} = \mathbf{0}$$

We finally get $\delta\boldsymbol{\theta}$:

$$\delta\boldsymbol{\theta} = -[J^T J + \lambda \operatorname{diag}(J^T J)]^{-1} J^T \mathbf{g}(\boldsymbol{\theta})$$

where $J = [J_1, J_2]$, $J_{i1} = \frac{\partial d(p_i)}{\partial \theta_1}$ and $J_{i2} = \frac{\partial d(p_i)}{\partial \theta_2}$ and $\mathbf{g}(\boldsymbol{\theta})$ is a column vector whose entries are $d(p, \boldsymbol{\theta})$ for each point p . λ is a damping factor set to 0.01 initially and adapting it throughout iterations.

In the following, we assume $0 < \tau_\alpha(p) < 1$ and $n_{\tilde{p}} \cdot n_p > 0$. In this case, p projects on \tilde{p}_α and $d(p) = \|pp_\alpha^*\| - r_\alpha(p)$ with $r_\alpha(p) = (1 - \tau_\alpha(p))r_1 + \tau_\alpha(p)r_2$, and $\tau_\alpha(p) = \frac{\|c_1 p_\alpha^*\|}{l}$. Hence:

$$\frac{\partial d(p)}{\partial \theta_1} = \frac{1}{\cos \alpha} \frac{\partial \|p^* p\|}{\partial \theta_1} + (r_2 - r_1) \frac{1}{l} \left(\frac{\partial \|c_1 p^*\|}{\partial \theta_1} + \tan \alpha \frac{\partial \|p^* p\|}{\partial \theta_1} \right) \quad (2.9)$$

$$\frac{\partial d(p)}{\partial \theta_2} = \frac{1}{\cos \alpha} \frac{\partial \|p^* p\|}{\partial \theta_2} + (r_2 - r_1) \frac{1}{l} \left(\frac{\partial \|c_1 p^*\|}{\partial \theta_2} + \tan \alpha \frac{\partial \|p^* p\|}{\partial \theta_2} \right) \quad (2.10)$$

The full expression for the derivatives can be easily derived given the expressions for $\|p_\alpha^* p\|$,

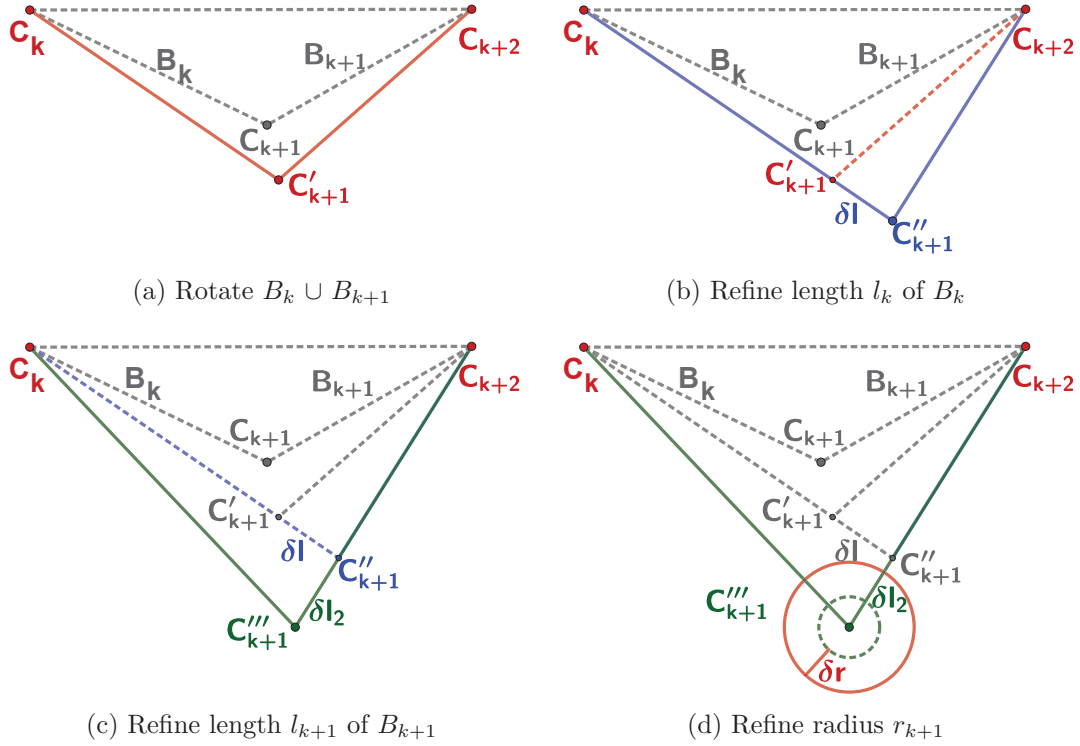


Figure 2.5 – Pairwise Optimization. With fixed extremities c_k and c_{k+2} , the pair of bones B_k and B_{k+1} is first rotated around axis $c_k c_{k+2}$ in order to minimize the two-bones energy. Then the lengths of the bones B_k and B_{k+1} and their common radius r_{k+1} are optimized successively. After these updates, the point-to-bone assignment is recomputed. As the process is repeated the distances are more accurate since the point-to-bone assignment becomes more meaningful.

$\|c_1 p^*\|$, $\|p^* p\|$ above. The cases $\tau_\alpha(p) < 0$, $\tau_\alpha(p) > 1$ or $n_{\hat{p}} \cdot n_p < 0$ can be computed similarly.

2.3.2 Optimization for a joint between two consecutive bones

The optimization of the position and radius of the joint between two consecutive bones (B_k, B_{k+1}) is performed by optimizing a set of four parameters in a loop (an angle, two lengths and a radius) minimizing the two-bones energy. The two end-sphere centers being fixed (c_k and c_{k+2} in Figure 2.5), we first compute the optimal rotation of the two bones around axis $c_k c_{k+2}$. We then optimize the bone lengths $\hat{l}_k = l_k + \delta l_k$ and $\hat{l}_{k+1} = l_{k+1} + \delta l_{k+1}$ and, finally, the radius of the common joint is computed as $\hat{r}_{k+1} = r_{k+1} + \delta r$. The parameters optimization alternates with a re-computation of point sets P_k and P_{k+1} , which refines the point-to-bone assignment. The optimization is also performed using the Levenberg-Marquardt algorithm.

Let us consider the geometric optimization of the joint's center between two bones by optimizing the two-bones energy concerning the lengths l_k and l_{k+1} . Each length is optimized in

turn, with a side-effect on the other length's value. The two-bones energy can be expressed as a function of l_k :

$$E_{(k,k+1)}(l_k) = \sum_{p \in P_k} \|\tilde{p}_k - p\|^2 + \sum_{p \in P_{k+1}} \|\tilde{p}_{k+1} - p\|^2 \quad (2.11)$$

Following the Levenberg-Marquardt algorithm, at each iteration, each parameter l_k is replaced by a new estimate $l_k + \delta l$:

$$\operatorname{argmin}_{l_k} E_{(k,k+1)}(l_k) \approx \operatorname{argmin}_{\delta l} E_{(k,k+1)}(l_k + \delta l) \quad (2.12)$$

By setting $\frac{\partial E_{(k,k+1)}(l_k + \delta l)}{\partial \delta l} = 0$, we get:

$$\delta l = - \frac{\sum_{p \in P_k} d_k \frac{\partial d_k}{\partial l_k} + \sum_{p \in P_{k+1}} d_{k+1} \frac{\partial d_{k+1}}{\partial l_k}}{\sum_{p \in P_k} \left(\frac{\partial d_k}{\partial l_k}\right)^2 + \sum_{p \in P_{k+1}} \left(\frac{\partial d_{k+1}}{\partial l_k}\right)^2} \quad (2.13)$$

where $d_k = \|p - \tilde{p}_k\|$ and $d_{k+1} = \|p - \tilde{p}_{k+1}\|$ are expressed as functions of l_k .

Let us detail the expression of d_k with respect to l_k : during the pairwise optimization c_k and c_{k+2} remain fixed (Figure 2.5). Let c_k be the origin of a local reference frame with the x-axis aligned with $c_k c_{k+1}$. In this frame, the coordinates write $c_k(0, 0, 0)$, $c_{k+1}(l_k, 0, 0)$ and $c_{k+2}(x_2, y_2, z_2)$ while a point P has coordinates (x, y, z) . Then $c_{k+1} c_{k+2} = (x_2 - l_k, y_2, z_2)$, $c_{k+1} p = (x - l_k, y, z)$.

Let us assume that p projects on \tilde{p}_α (the case $\tilde{p}_{-\alpha}$ can be deduced with minor changes). Using the same notation as in Figure 2.4 and section 2.3.1, recall that $d_k = \|p - \tilde{p}_k\| = \|pp^*\| - r_\alpha(p)$. Since when optimizing l_k the orthogonal projection on $c_k c_{k+1}$ does not change, $\|pp^*\|$ remains the same. However both α and $r_\alpha(p)$ change. Since $r_\alpha(p) = (1 - \tau_\alpha(p))r_k + \tau_\alpha(p)r_{k+1}$ with $\tau_\alpha(p) = \frac{\|c_k p^*\|}{l_k}$, we get:

$$\frac{\partial d_k}{\partial l_k} = - \frac{\|pp^*\|}{\cos^2 \alpha} \frac{\partial \cos \alpha}{\partial l_k} - (r_{k+1} - r_k) \frac{\partial \tau_\alpha(p)}{\partial l_k} \quad (2.14)$$

Simple geometric considerations give $\cos \alpha = \sqrt{1 - \frac{(r_{k+1} - r_k)^2}{l_k^2}}$, $\tau_\alpha(p) = \frac{\|c_k p^*\| + \|pp^*\| \tan \alpha}{l_k}$ and $\tan \alpha = \frac{r_{k+1} - r_k}{\sqrt{l_k^2 - (r_{k+1} - r_k)^2}}$, whose differentiation with respect to l_k is easy.

One must also express distances d_{k+1} as functions of l_k . In that case, the projection on bone B_{k+1} is slightly different, since the position of point c_{k+1} changes with l_k . The formulas are only slightly modified by it, but this time $\|pp_\star^*\|$ also depends on l_k . We get:

$$\frac{\partial d_{k+1}}{\partial l_k} = \frac{1}{\cos \alpha} \frac{\partial \|pp_\star^*\|}{\partial l_k} - \frac{\|pp_\star^*\|}{\cos^2 \alpha} \frac{\partial \cos \alpha}{\partial l_k} - (r_{k+2} - r_{k+1}) \frac{\partial \tau_\alpha}{\partial l_k} \quad (2.15)$$

The full expression for the derivatives can be easily computed using the following formulas:

$$\cos \alpha = \sqrt{1 - \frac{(r_{k+2} - r_{k+1})^2}{(x_2 - l_k)^2 + y_2^2 + z_2^2}}$$

$$\tau_\alpha(p) = \sqrt{\frac{(x - l_k)^2 + y^2 + z^2}{(x_2 - l_k)^2 + y_2^2 + z_2^2}}$$

$$\|c_{k+1}p^*\| = \frac{c_{k+1}p \cdot c_{k+1}c_{k+2}}{\|c_{k+1}c_{k+2}\|} = \frac{(x - l_k)(x_2 - l_k) + yy_2 + zz_2}{(x_2 - l_k)^2 + y_2^2 + z_2^2}$$

Plugging all the derivatives in Equation 2.13 yields δl , and l_k can be updated as $\hat{l}_k = l_k + \delta l$. This impacts the position of c_{k+1} , whose new position is computed as $\hat{c}_{k+1} = c_k + \hat{l}_k \frac{c_k c_{k+1}}{c_k c_{k+1}}$, and l_{k+1} is recomputed as : $l_{k+1} = \|\hat{c}_{k+1}c_{k+2}\|$.

The two-bones energy $E_{k,k+1}$ is then optimized with respect to l_{k+1} . This optimization is symmetric to the l_k case above and can be easily adapted. Finally, the optimization of the radius of the common joint and rotation angle around axis $c_k c_{k+2}$ is done similarly.

2.4 FAKIR : Forward And bacKward Iterative Registration

To register our anatomical model, we propose a kinematic approach considering the way the model is articulated. Contrarily to many methods that work from videos or multiple views [TPT16, RTTP17], our method requires only one joint center to be close to its optimal position, the rest of the skeleton pose being arbitrary. Inspired by the FABRIK [AL11] and CCD [WC91] algorithms, our registration algorithm successively loops forward and backward through the chains of bones to rotate and scale them to match the data, refining the parameters while temporarily fixing the extremities of some bones. Hence our algorithm is named Forward And bacKward Iterative Registration (FAKIR). Our method's originality is that bones are mainly considered by consecutive pairs, which allows for a more robust estimation of the pose and skeleton parameters along a chain. The optimization of parameters related to bone B_k requires that relevant points attract that bone into the data attachment term, which justifies a special order for the optimizations.

2.4.1 Registration process for a chain of bones.

If parameters of bone B_k have not yet been initialized, and B_k is close to a subset of points to which it should ideally associate, the estimation of the one-bone energy is meaningful, and the minimization of this energy can be used to initialize the position and radii of that bone concerning the data. Our algorithm gradually rotates and scales the current bone B_k with respect to its predecessor, updating P_k after each step, so that P_k gradually contains more relevant points. However, if P_k is empty, the bone is first rotated around the three axes until

some input points are projected onto it to bootstrap the optimization. Once the position of B_k has been approximately found, the algorithm turns to the coarse estimation of the position of B_{k+1} . All these computations are driven by the minimization of the one-bone energy. However, the one-bone energy alone might be inefficient to accurately approximate the full length of a bone. To alleviate that, in an intertwined manner, a finer local registration is performed each time two consecutive bones B_k and B_{k+1} have been processed, by minimizing the two-bone energy. This process optimizes the common joint position and radius while keeping the two other joints fixed. It is the essential component in our algorithm. Once a chain of K bones has been positioned and scaled over its entire length, we repeat the process forward and backward in the chain to further refine the joints positions and radii between pairs of consecutive bones, using only two-bone energies optimizations. Extremity bones are optimized based on the one-bone energy after each forward or backward pass. The full process is summarized in Algorithm 1 and illustrated with a chain of three bones in Figure 2.6.

Notice that if two limbs are aligned, the joint position can not be guessed from the data and it may cause several limbs to be included in a single primitive of our sphere-mesh. To avoid this, very loose constraints on the differences in proportions between consecutive bones can be set.

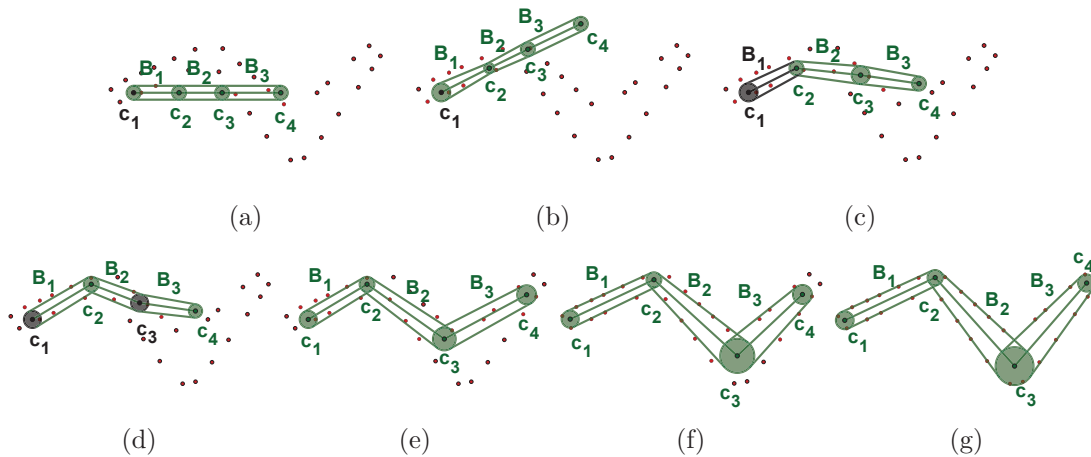


Figure 2.6 – Overview of the forward and backward iterative registration for a 3-bone chain. From an initial position (a), the chain extremity c_1 is fixed and the first bone B_1 is rotated and scaled to roughly calibrate its dimensions and pose through the optimization of the one-bone energy (b); the bone B_1 is fixed and the parameters of the second bone B_2 are roughly calibrated in turn (c); joint c_2 which is common to the first two bones is scaled and its position is optimized, by using the two-bones energy, the other joints being fixed (d); The position and length of the third bone B_3 are then coarsely calibrated through one-bone optimization and the process continues by alternating single bone optimization and two-bones optimization, until the last bone of the chain (e). After this first coarse calibration forward pass finishes, a backward pass using only two-bone optimizations is performed (f) permitting to refine the pose and skeleton parameters and solve for the chain extremity position. With few forward and backward pass involving two-bone optimization only, the model is registered (g).

Algorithm 1 Forward and backward iterative registration

Input: A point set P and a sphere-mesh chain of K bones with one chain extremity close to its optimal position

Output: The registered sphere-mesh chain.

- 1: **Initialization:**
 - 2: Fix the center of the first extremity of the chain. Rotate the first bone and adjust its radii and length by minimizing the one-bone energy;
 - 3: **for** $k := 1$ **to** $K - 1$ **do**
 - 4: Consider the pair of bones B_k, B_{k+1} :
 - 5: Fix the position of the joint common to B_k and B_{k+1} ;
 - 6: Alternate between the optimization of B_{k+1} 's rotation w.r.t B_k , optimization of B_{k+1} 's intrinsic parameters and update of P_{k+1} ;
 - 7: Fix the positions of the 2 joints that B_k and B_{k+1} do not share, and free their common joint;
 - 8: Compute the position and the radius of the common joint by using the two-bones energy.
 - 9: **end for**
 - 10: Compute the length of the last bone and the radius of the last sphere.
 - 11: **Forward and Backward registration loop:**
 - 12: **repeat**
 - 13: Reverse the order of the bones in the chain;
 - 14: **for** $k := 1$ **to** $K - 1$ **do**
 - 15: Consider the pair of bones B_k, B_{k+1} :
 - 16: Fix the positions of the 2 joints that B_k and B_{k+1} do not share;
 - 17: Compute the position and the radius of the common joint by using the two-bones energy.
 - 18: **end for**
 - 19: Compute the length of the last bone and the radius of the last sphere with the one-bone energy.
 - 20: **until** convergence
-

2.4.2 Full Skeleton Registration

The full model corresponds to a tree whose branches are composed of chains. The process of registering each of the chains must be done to gradually ensure the relevance of the data attached to each chain. Thus, previously registered chains can be questioned again if their attached points are reassigned to other chains or if they catch new points during the process (see Figure 2.7).

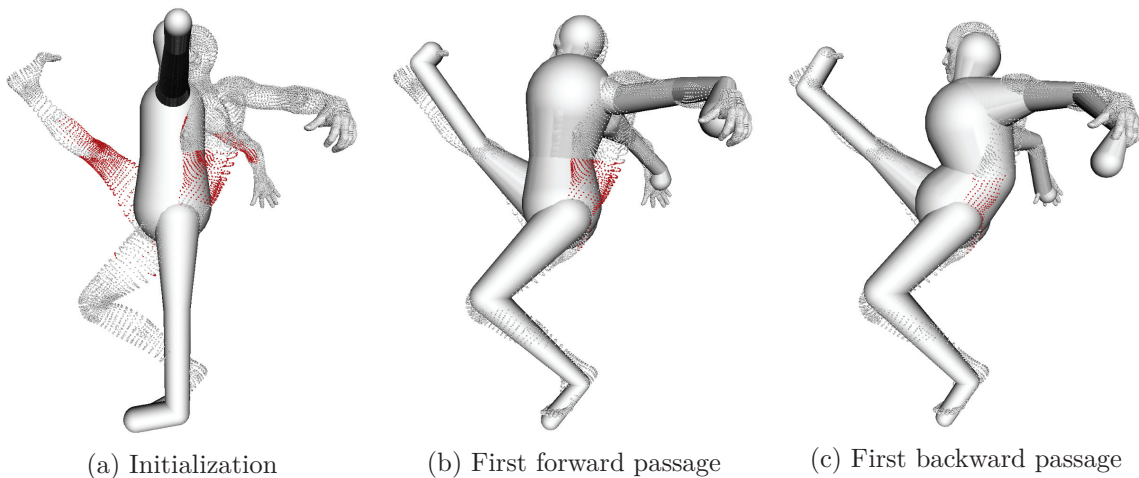


Figure 2.7 – The assigned points for the first spine is showed in red. The spine is first registered with points that are not yet assigned to the legs, which distorts its position. Once the legs are registered, the registration backtracks to the spine and its position can be corrected. The final result is shown on Figure 2.16.

Our system depends on the initial position of a joint chosen as the skeleton’s root. For example, we assume that the pelvis part of our model is initialized near the corresponding part of the point set for human or quadrupeds models, which is done manually through a single point and click. Each chain is then registered in turn using FAKIR yielding a registered skeleton both in terms of intrinsic parameters and pose in only a couple of iterations.

For human models, the registration order is the following: first, the spine chain is registered, refining the pelvis position and scale during the process, followed by each of the two leg chains and each of the two arm chains. When registering the arms and legs chains, the joint’s position attached to the spine or the pelvis remains fixed. However, after one arm is registered, changing the spine-arm joint position, the spine chain is updated accordingly (and similarly for legs and head). Thus the forward and backward chain registration extends to the whole model. This leads to a calibrated and accurately positioned articulated model.

2.5 Results

In this section, we show the performance of FAKIR both on synthetic data and on point sets resulting from statue digitization. We developed our algorithm in C++, using OpenMP for computing point to bone distances in parallel. All experiments are run on an Intel Core i7-4790K CPU @ 4.00GHz. Normals were computed by using the state of the art approach of Hoppe et al [HDD*92].

2.5.1 Experiments on synthetic data

We first tested our algorithm on synthetic data to provide a quantitative evaluation of the FAKIR performances. We considered a point set of $5k$ points sampled on a sphere-mesh of a 4-bone chain in a specific pose and registered a generic 4-bone chain to it. Although the point set and the initial chain are quite distant from each other, providing an approximate initial position of a single anchor point (one of the extremity) is enough to register accurately the chain. The accuracy of the registration is evaluated as the average distance between the point set and the model.

$$dist = \frac{1}{N_{points}} \sum_{p \in P} \|p - \tilde{p}\|. \quad (2.16)$$

In the noiseless case, our algorithm takes 7 iterations to converge to a 0 distance in $2.37s$, including $0.62s$ for the first forward pass. The distance of the point set to the model with respect to the iterations for larger point sets and increasing noise is shown on Figure 2.8: the number of points has only a moderate impact on the number of iterations needed to converge (around 7). When there is noise in the data, the distance also converges in a few iterations independently of the noise, however the distance at convergence is directly correlated to the variance of the noise. As shown by our experiments, FAKIR is rather resilient to even relatively high levels of noise (Gaussian noise in Figure 2.9 and Poisson noise in Figure 2.10). Figure 2.11 shows how FAKIR handles an initial position of the anchor point that is not in the vicinity of its optimal position in the point set. FAKIR can handle initial positions that are moderately far from the true position, but in some cases (last column), the backward optimization of the one-bone energy alone fails to reduce the length of the first bone and the radius of its free extremity degenerates to 0 instead. This is due to the fact that no point is projected on the spherical free extremity of that bone. This problem could be avoided by adding a bone occupancy term to the one-bone energy. A preferential alternative would be to modify the one-bone energy of the first and last bones by adding a term corresponding to the distance of the free caps to the data points. However, if the initial point is reasonably close to its true position, this problem does not occur. FAKIR is also rather robust to missing data thanks to the iterated forward and backward passes (Figure 2.12). Naturally when the missing parts are on the first or last bone or when a full bone is missing,

the algorithm cannot predict the right length or angle.

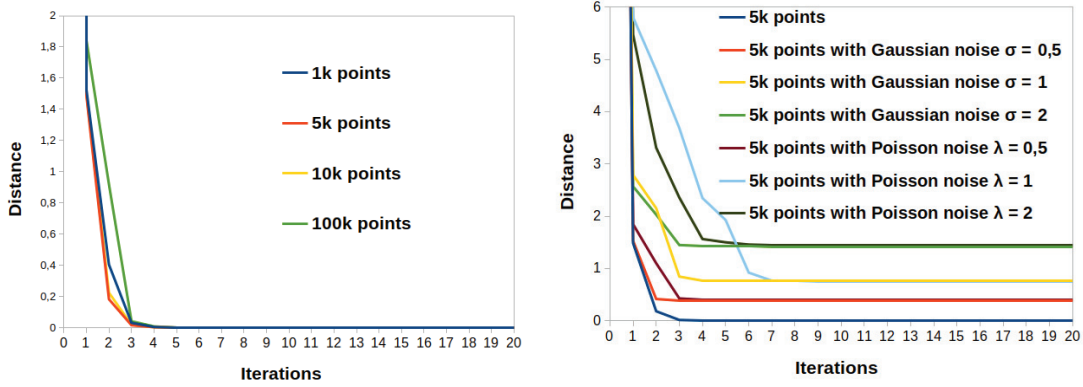


Figure 2.8 – Evolution of the registration distance with the iterations for different number of points in the point set (left image), different levels of Gaussian noise and Poisson noise (right image).

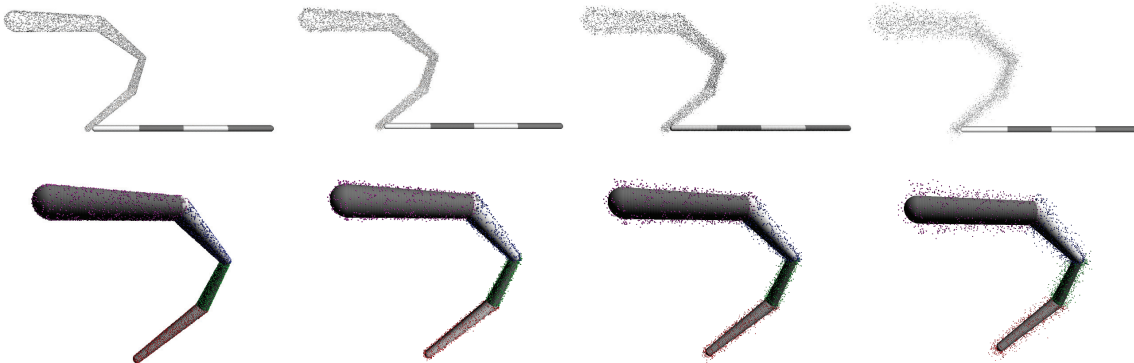


Figure 2.9 – Evaluation of FAKIR with respect to increasing Gaussian noise after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain. From left to right: without noise, $\sigma = 0.5$, $\sigma = 1$, and $\sigma = 2$. The total groundtruth model length is 140 (All values are given in length units).

2.5.2 Skeleton registration results on statues

We selected some interesting statues from various sources.

Complete models:

1. Dancer with Crotales, Louvre Museum
2. The Goddess Parvati, South India
3. Saint John the Baptist, Ny Carlsberg Glyptotek

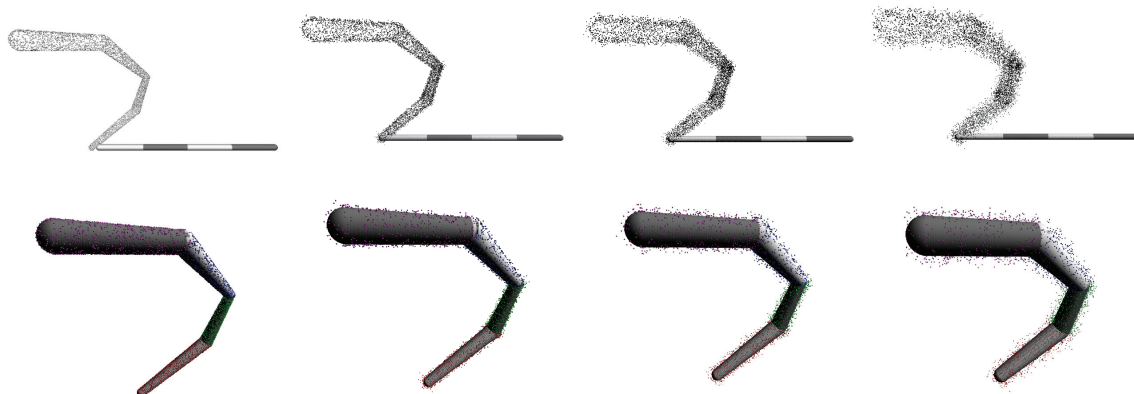


Figure 2.10 – Evaluation of FAKIR with respect to increasing Poisson noise after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain. From left to right: without noise, $\lambda = 0.5$, $\lambda = 1$, and $\lambda = 2$. The total groundtruth model length is 140 (All values are given in length units).

4. Aphrodite, Thorvaldsens Museum
5. Dancing Faun, Pompei excavations
6. Age of Bronze,
7. Mermaid, Royal Bibliotek of Copenhagen

Incomplete models:

1. A plaster copy of Esquiline Venus, The Royal Cast Collection
2. A copy of The Old Fisherman, The Royal Cast Collection
3. A statue in the style of the Venus De Milo
4. Wounded amazon, Ny Carlsberg Glyptotek

While the 'Dancer with crotales' is a raw point set. The other 7 models are point sets sampled on meshes extracted from the Sketchfab website. We also use several models from the TOSCA dataset [BBK08], including nonhuman models to show the pliability of our method.

Figure 3.9 shows our registrations on four statues. The registration algorithm performs well for statues depicting naked characters: in this case, the registration is not hindered by additional clothing or accessories, and the simple sphere-mesh model fits well the data. Even with moderate clothing (Dancer with Crotales) FAKIR recovers the pose of the statue. FAKIR can also work on incomplete statues on as shown in Figure 2.14 and on imaginary creature statues (Figure 2.15). We also demonstrate that FAKIR can work with real human bodies in more complex poses (Figure 2.16) or animals (Figure 2.17) from the TOSCA dataset.

While skeleton extraction is a much explored topic in geometry processing, we emphasize that extracting a computational geometry skeleton is very different than extracting an anatomical

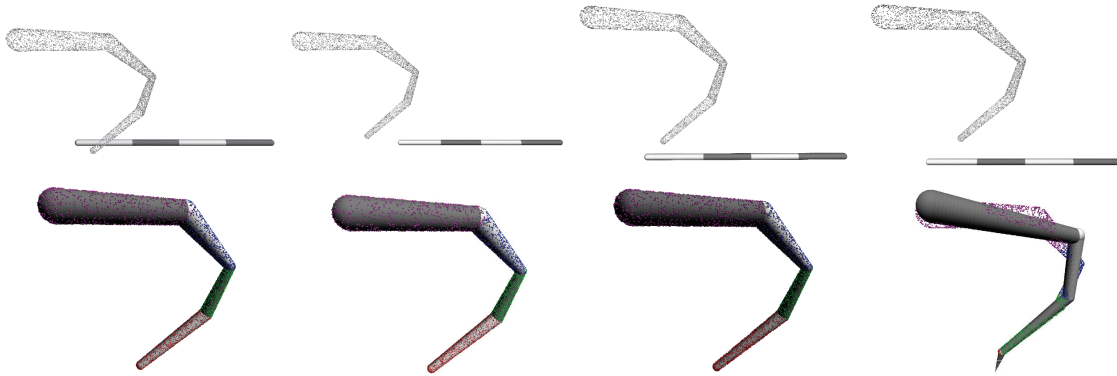


Figure 2.11 – Evaluation of FAKIR with respect to a bad initial anchor point position after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain. The last column shows that due to a bad initialization, the points (plotted in red) that are affected to the first bone do not bring enough information for the one-bone energy to move the chain extremity. Then, not enough bones remain to approximate the whole point set.

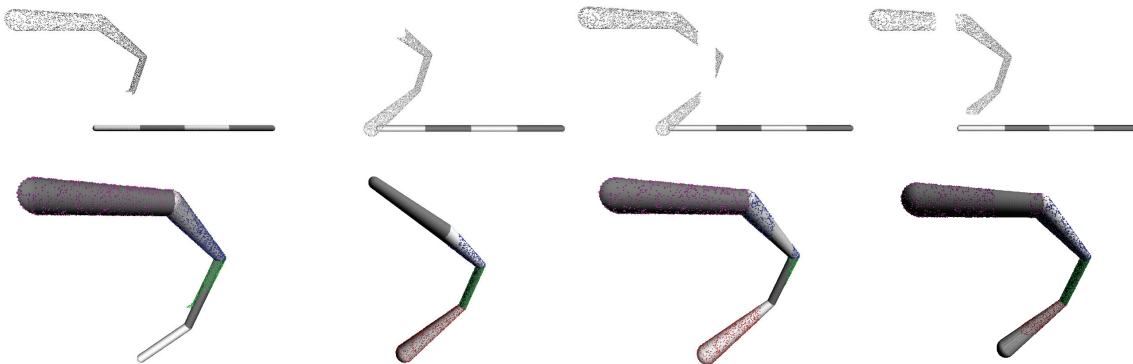


Figure 2.12 – Evaluation of the FAKIR algorithm with respect to missing data after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain.

skeleton [HWCO*13, TDS*16]. Figure 2.18 show the ℓ^1 -medial skeleton extracted [HWCO*13] on the Aphrodite and Danseuse with Crotales point sets – to be compared with our results on Figures 2.19 and 2.20. This experiment shows that the geometrical skeleton definition is not enough for our purpose.

We compare FAKIR with Pinocchio [BP07] in Figure 2.19. The FAKIR algorithm yields a better skeleton registration, in particular for the shoulders and neck bones. As far as computation times are concerned, the Pinocchio method takes about 35s for a mesh with 138048 vertices, which is roughly the same time as the 10 iterations of FAKIR optimizing not only for the joint positions but also for the bone radii (38s). Furthermore, a single iteration of FAKIR takes 9s and already provides a better result with a much more plausible shoulders location. However it is important to note that the Pinocchio method does not require an initial skeleton position, while

our method requires one of the joint to be not far from its optimal position (in this experiment we chose the pelvis joint).

We also compare FAKIR with the SMPLify method [BBLR15] (Figure 2.20). First we compute a front-view rendering of the shape and run DeepCut [PIT*16] to estimate the joint position used for SMPLify. The registration is clearly less accurate than ours (Figure 2.20b). We then extend SMPLify method to multi-view images using epipolar constraints to estimate a 3D joint positions from 2D joint positions obtained by DeepCut before applying SMPLify, which slightly improves the registration (Figure 2.20c). We also show that our articulated model regression can serve to initialize an SMPL model, leading to a better registration than with DeepCut (Figure 2.20d). However, the shape estimation of SMPL still cannot fit a statue with non-realistic body proportions.

Finally we compare FAKIR with the FARM method [MMRC20] a fully automatic method for registering an underlying model, such as SMPL, to a point set or a mesh. While this method exhibits excellent results on humanoid shapes, in the case of artistic statues, with unrealistic body proportions (Figure 2.19) or with moderate garments (Figure 2.20), it cannot reconstruct a plausible model. Furthermore, scanned statues often exhibit inconsistent topologies, such as an arm glued to the body (Figure 2.19) which is not handled by the FARM method.

2.5.3 Computation time

The computational bottleneck of FAKIR lies in the assignment of each point several times during the optimization process. This assignment is updated after each bone parameter change. However, the number of updates is related to the geometry of the surface and not to the number of sample points. Therefore, the overall complexity is linear with respect to the number of points. From an experimental point of view, FAKIR is a reasonably light algorithm: for a point cloud of 10000 points and the 22-bone human model, the first forward pass of FAKIR takes 2.5s and the computation time for one pass decreases to 1s in average afterwards.

2.6 Discussion

2.6.1 Limitations

Despite its good results, FAKIR has some limitations. First, if two consecutive bones are aligned, their length estimation is not reliable, since the position of the middle joint is undetermined if no proportion constraint is set on the model. This limitation appears on the Aphrodite left leg (Figure 3.9). We could also improve registration results by adding constraints of symmetry to the energy functions. However these constraints should be quite loose, because of the unrealistic proportions of artistic statues. Furthermore, failure cases include some misalignments due to a local minimum (one of the arms of the mermaid in Figure 2.15, the feet of the shape

on the top-right of Figure 2.16 and the back leg of the centaur in Figure 2.17.) Further local extrinsic refinement would improve the result. In addition, a process of local modification of the skeleton structure could be carried on, followed by an update of its registration. Last, to avoid any manual intervention, we have tested an automatic approach by initializing the pelvis in the center of the bounding box of the points and by orienting it upwards. This initialization is effective for most standing statues (results provided in the section 2.6.4). Using principal component analysis to initialize the orientation of the body would also improve the registration for classical human poses (e.g. walking, sitting, lying down). However, the handling of the relaxation process at bifurcations could also be improved. Currently, the position of the pelvis is not updated by exploiting the whole information arising from the registration of all the incident branches.

2.6.2 Importance of the optimization order for registering a chain of bones.

During our optimization process our approach takes advantage of the articulated property of our model by processing bones in a specific order. Here, we run an experiment to illustrate that the order in which the optimizations are made is crucial.

To do so, we replace our iterations of sequential optimizations followed by point to bone reassignment by iterations of simultaneous parameter optimizations followed by point reassignment. At each iteration, the positions of all the joints are simultaneously optimized by minimizing their two-bone energies as if the adjacent joints remained fixed. The free joints at the extremities are also optimized at the same time by minimizing their one-bone energy as if their non free joint remaining fixed. However the parameter and pose change is not applied right away after each optimization but simultaneously once all updates have been computed.

If this optimization is run after our forward step (which is useful to bring each bone close to relevant data), it takes 43 iterations to converge, against 9 iterations only with our approach (See Figure 2.21 for an illustration of the stages). On the contrary, if the simultaneous optimization is run directly from the initial position, the method fails to converge. Figure 2.22 shows the result after 50 simultaneous optimizations steps.

2.6.3 Importance of the normal-constrained projection

In this section we demonstrate that the normal-constrained projection both improves the result of the registration and the computation time. It is especially true in the case of the Aphrodite statue (Figure 2.23). Indeed, for this statue, the arms cling to the body which leads to wrong assignment of points when no normal information is used, yielding an unrealistic statue pose. The normal-constrained projection, on the contrary, permits to recover a good pose of the arms. Furthermore, it takes 16.9s and 2 iterations for the algorithm using normal-constrained projection, against 32.8s and 8 iterations – to converge to a wrong registration – otherwise (number of points: 38954). On simpler cases, like the Dancing Faun (see the main paper for a

rendering of the shape), both methods manage to converge to the correct registration, but the computation time and number of iterations are still lower for the normal-constrained projection (8.7s, 6 iterations) than for the simple orthogonal projection (10.4s, 10 iterations).

2.6.4 Automatic initialization test

Our attempt to automatically initialize FAKIR by placing the pelvis in the center of the bounding box is effective for many statues. The Figure 2.24 illustrates that this works with a vertically oriented statue, while the Figure 2.25 shows a failure result, due to the fact that the center of the bounding box does not give any information about the orientation of the body. Thus the chains of bones are not aligned with the appropriate points, which leads to a local minimum. An improved version using principal component analysis of the points to initialize the orientation of the pelvis and adding loose constraints on the length of the bones could improve the registration.

2.7 Conclusion and perspectives

We introduced a sphere-mesh anatomical model and a combined calibration and registration algorithm to estimate the anatomy and the pose of digitized archaeological statues. Our algorithm is useful when it is not possible to extract a shape template from a statistical analysis of examples representative of the diversity of poses and morphologies. Given the simplicity of our anatomical model, it is very easy to adapt it to many shapes, such as animals or imaginary creatures. While our method already gives good results, a further improvement would be to handle the case of a clothed statue.

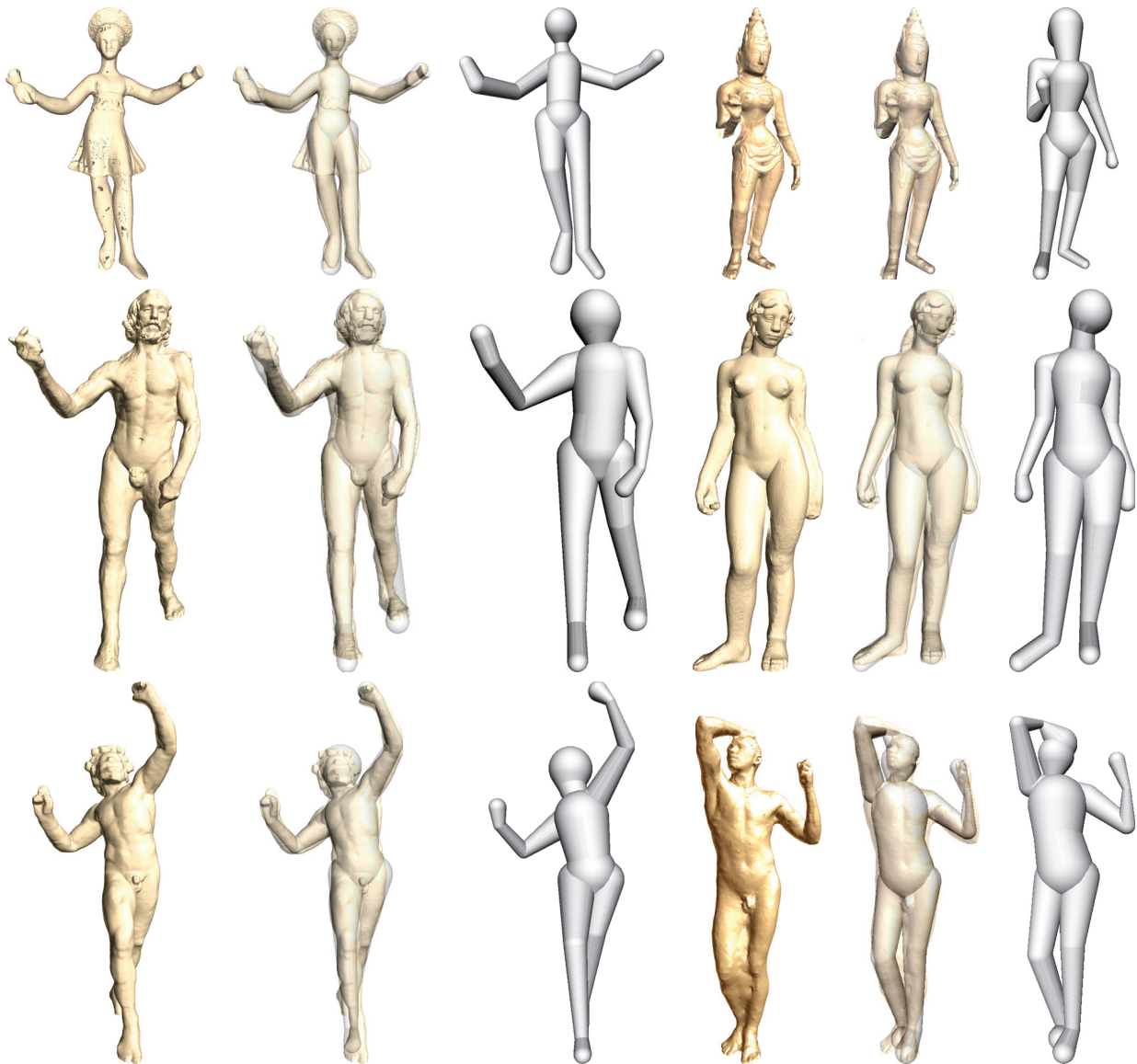
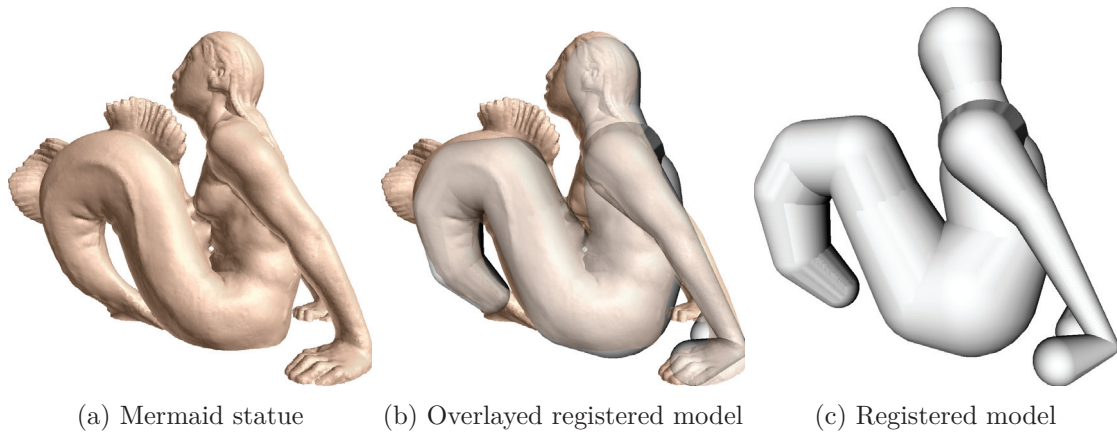


Figure 2.13 – Registration of 6 statues: the Dancer with Crotales (first row at left), the Goddess Parvati (first row at right), the Saint John the Baptist (second row at left), the Aphrodite (second row at right), the Dancing Faun (third row at left) and the Age of Bronze (third row at right). For each result, we show from left to right: initial point set, overlay of the registered model and the point cloud, registered model.



Figure 2.14 – Registration of 4 incomplete statues, Esquiline Venus (first row), Old Fisherman (second row), Venus de Milo (third row) and Wounded Amazon (fourth row). First column: initial point set, second column: overlay of the registered model and the point set, third column: registered model alone.



(a) Mermaid statue

(b) Overlaid registered model

(c) Registered model

Figure 2.15 – FAKIR result on the Mermaid statue, the skeleton is a human one but with a single chain instead of two legs.

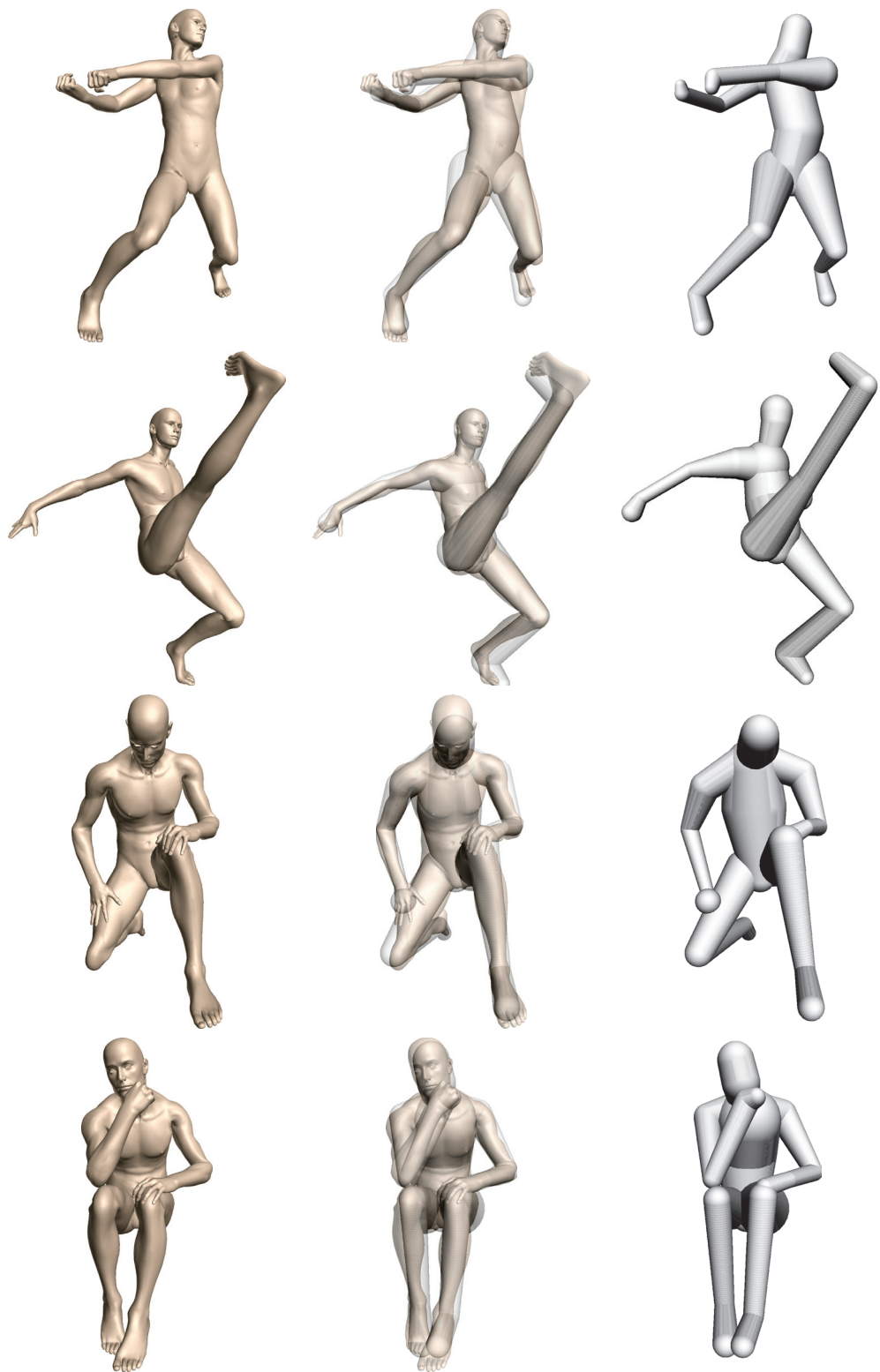


Figure 2.16 – Skeleton registration on human in various poses from the TOSCA dataset.

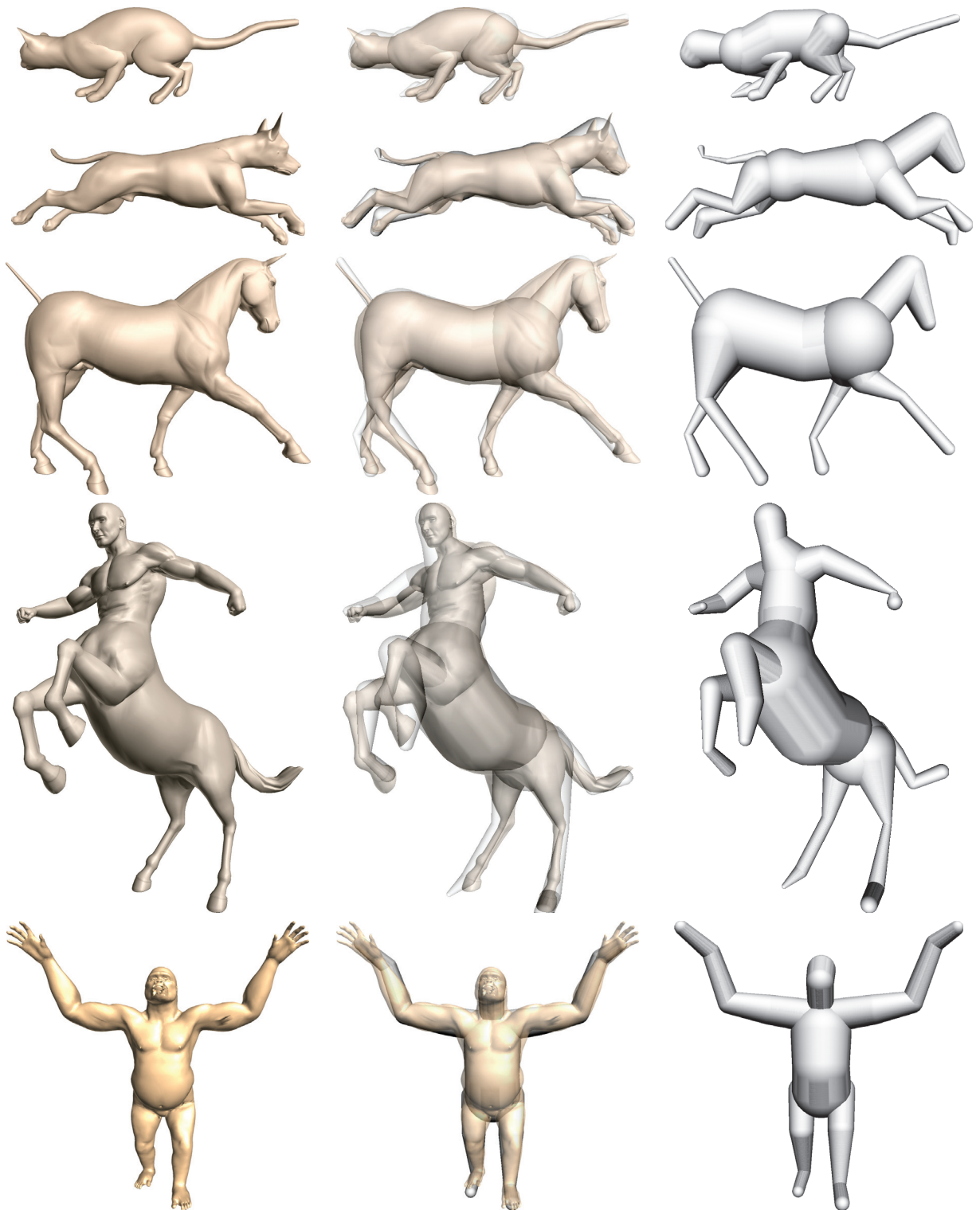


Figure 2.17 – Skeleton registration on different animals or created characters from the TOSCA dataset. The skeleton is simply the human skeleton (Figure 2.3) supplemented with a bone chain for the tail.

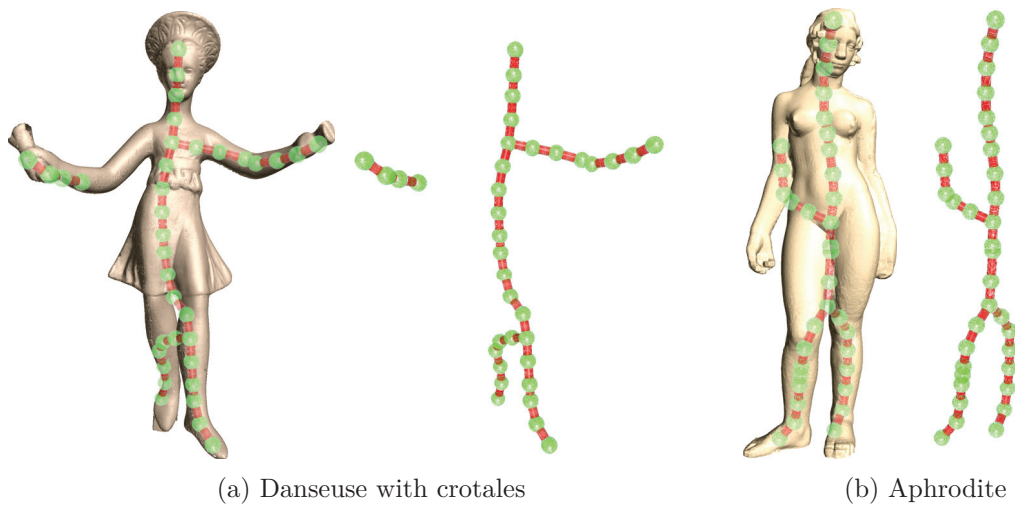


Figure 2.18 – ℓ^1 -medial skeletons [HWCO*13] extracted from the Aphrodite and Danseuse with Crotales point sets. To compare to our results on Figures 2.19 and 2.20.

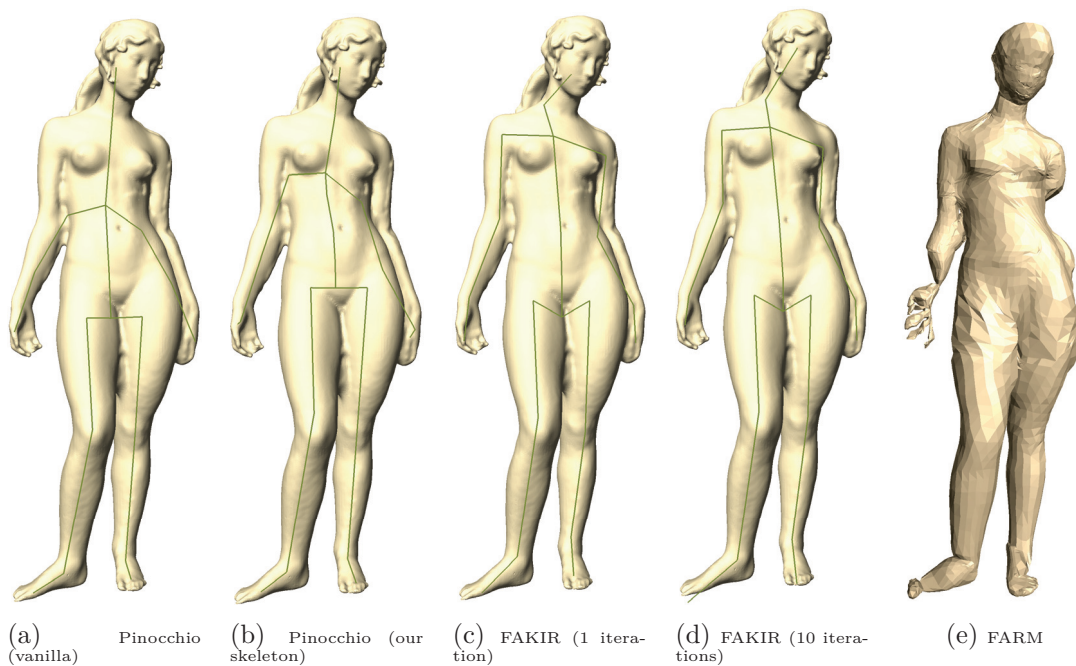
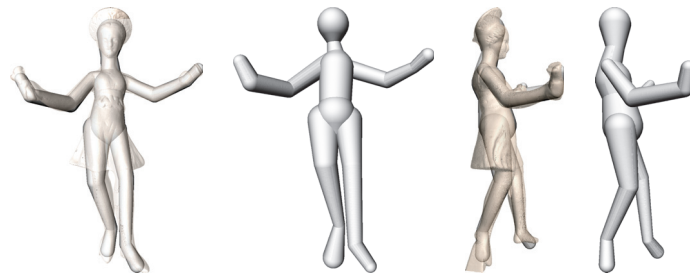
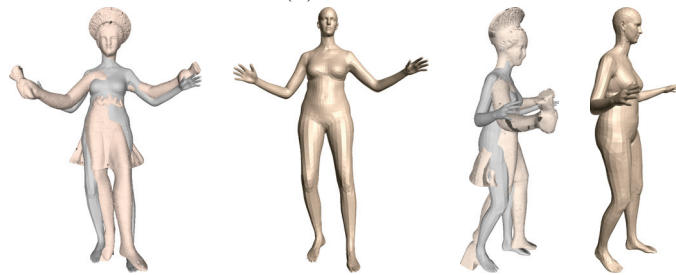


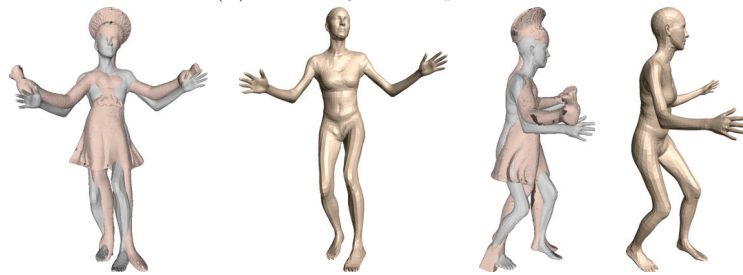
Figure 2.19 – Comparison with Pinocchio [BP07] and FARM [MMRC20] algorithm on the Aphrodite statue. From left to right: (a) Pinocchio with the Pinocchio-provided initial skeleton (17 bones); (b) Pinocchio with our initial skeleton (22 bones); (c) FAKIR with our initial skeleton after a single forward iteration; (d) FAKIR with our initial skeleton in 10 iterations. Only the skeleton is displayed since the bone radii are not taken into account by Pinocchio. (e) Model reconstructed by the FARM method.



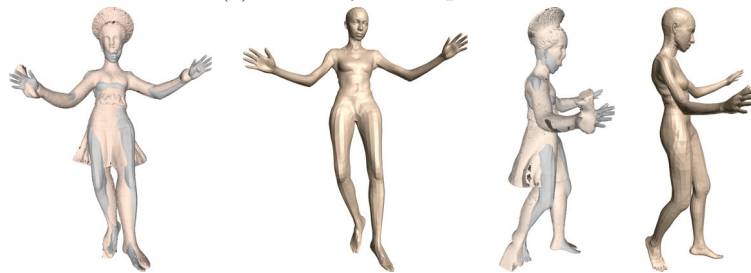
(a) FAKIR



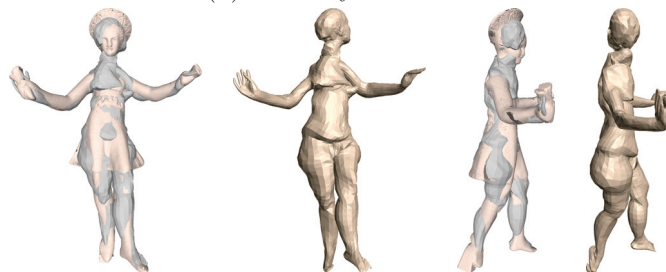
(b) SMPLify + DeepCut 2D



(c) SMPLify + DeepCut 3D



(d) SMPLify + FAKIR

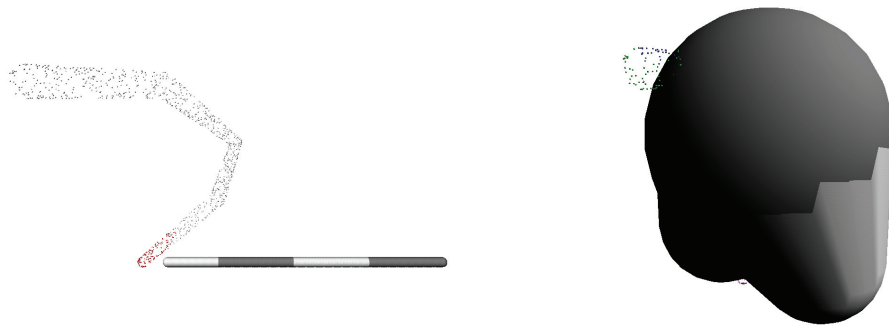


(e) FARM

Figure 2.20 – Comparison with SMPLify on the Danseuse with Crotales. (a) FAKIR registration; (b) SMPLify using DeepCut predicted 3D joint positions on a single rendering; (c) SMPLify using DeepCut predicted 3D joint positions on two rendered views; (d) SMPLify using 3D joint positions estimated by FAKIR; (e) FARM registration. (First and third column: overlaid registered model; Second and fourth column: registered model)



Figure 2.21 – Simultaneous Optimization for a chain of bones applied after the first Fakir forward pass. From left to right: initial position (after our forward step), position after 10 iterations, position after 35 iterations, position after convergence (43 iterations).



(a) Initial position for a simultaneous optimization

(b) Position after 50 iterations

Figure 2.22 – Simultaneous Optimization for a chain of bones applied directly from the initial position.

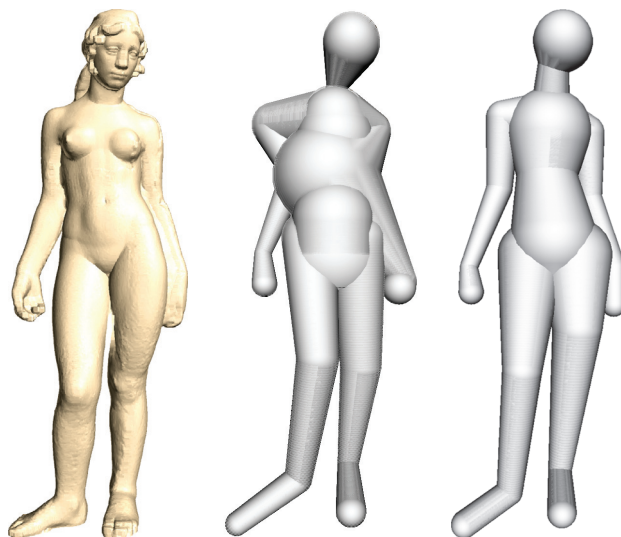


Figure 2.23 – From left to right: Aphrodite statue, registration result without the normal-constrained projection, registration result with our normal-constrained projection

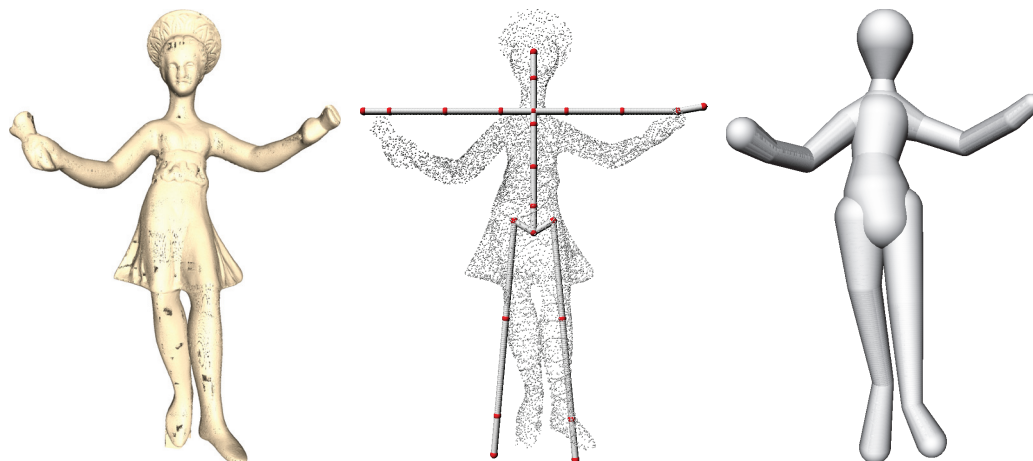


Figure 2.24 – From left to right: the Dancer with Crotales statue, initialization position at bounding box center of point cloud, automatic registration result.

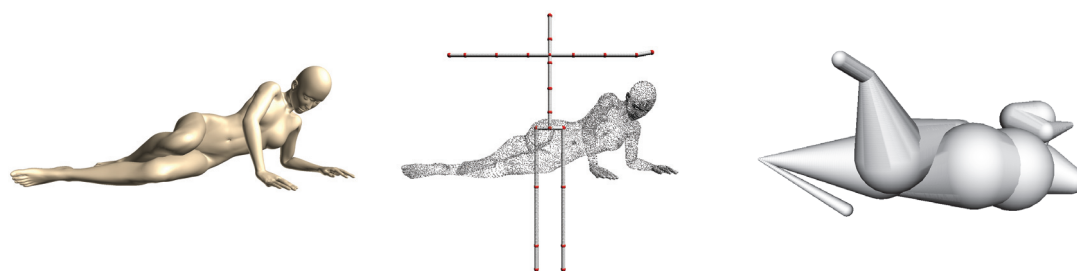


Figure 2.25 – A failure case for automatic initialization. From left to right: the Victoria from TOSCA data set, initialization position at bounding box center of point cloud, automatic registration result.

A FIRST APPROACH TO SPHERE-MESH MODEL SKINNING

Once the anatomical model is registered to the statue point set, we exploit the relative position of the points with respect to the sphere-mesh model to change the pose and the elementary anatomy of a statue by modifying the extrinsic and intrinsic parameters of the bones respectively and by reporting the point set details above it. One of the challenges when changing the pose and anatomy of a virtual statue is the production of plausible skin deformation at joint. Methods to give a new skin of a virtual character after a deformation are called skinning. The main idea of mesh skinning is to deform a mesh following an underlying skeletal animation. Each vertex is influenced by one or more joints and vertex weights describe the influence of relevant joints on each vertex.

Compared to existing animation approaches based on sphere-meshes, we work directly on the point set and not on a lower-resolution mesh to which a skinning algorithm can be applied after carefully setting weights. Using skinning on a mesh with fixed connectivity leads anyway to triangle slivers and possibly self-intersections and dropping the connectivity alleviates this problem. However, we have not given up the idea of using a skinning algorithm. It is applied directly to the base-points of the sphere-mesh model with weights that are independent of the detail and that do not need to be completely revised from one model to another.

In our case the skinning is applied to the set of base-points of the sphere-mesh model with adapted weights that only depend on the base-points. To do so, we need to attach the detail point set to its registered articulated model so that deforming the model will deform the point set accordingly. We propose to associate a base-point on the articulated model to each of the original points, the originality of our approach being that those base-points may locally slide on the moving sphere-mesh to mimic the sliding of the skin on joints. For instance, some base-points can evolve from the conic section of a bone to the neighboring sphere, resulting in continuous deformation with better volume preservation than the usual linear blend skinning. In addition, special care is given to the twisting and bending rotations at each joint. The new position of the input points is then deduced from the new position of their corresponding base-points, by reporting their initial height. We will explain this heightfield skinning in section 3.1. For moving

the base-points, we propose a new point set skinning method where the deformation at a joint is the combination of a twisting rotation and a bending rotation in section 3.2.

3.1 Heightfield skinning

3.1.1 Heightfield over the sphere-mesh model

We propose to consider the skin of model as a set of detail-points, encoded as a set of heights on top of our articulated model, that will be transferred back on the model after pose or intrinsic parameters change. This approach is in line with displacement maps on top of polygonal models [POC05], but instead of defining a complete heightfield over the model, we define it only on base-points. The same base-point can correspond to several detail points, making it possible to handle layered details, *i.e.* folded surface sheets over the sphere-mesh model (see the Dancer with Crotales on Figure 3.2 and the dragon on Figure 3.1).

In the following, let us assume that the sphere-mesh model is registered to the point set. The heightfield value h of a point p is defined as a signed distance from p to its projection on the sphere-mesh base point \tilde{p} : $p = \tilde{p} + h(p)\mathbf{n}_{\tilde{p}}$, where $\mathbf{n}_{\tilde{p}}$ is the normal to the sphere-mesh surface at \tilde{p} and \tilde{p} is the orthogonal projection on the bone p is assigned to. Hence the surface point set is decomposed into the set of base-points on the sphere-mesh and a residual orthogonal heightfield.

Using that decomposition, any skinning algorithm can be transformed into a heightfield skinning algorithm above a sphere-mesh. The base-points are skinned using the given algorithm and then the heightfield is added back to the modified base-points yielding the final point set (Algorithm 2). It is preferential to use a skinning algorithm where base-points remain on the sphere-mesh, otherwise we re-project them on the sphere-mesh, similarly to what is done with implicit skinning [VBG*13].

3.1.2 Continuity

The positions of detail points continuously evolve with the parameters of sphere-mesh and there is no tearing of the surface. Indeed, a base-point can slide continuously along a bone and move from the conic part to the sphere part of the bone and conversely. Since the conic part is tangent to the sphere, both the base-point and its normal evolve continuously even in case of cone to sphere or sphere to cone base point motion. Since the heightfield of a detail-point is either preserved or continuously scaled, the whole motion is continuous throughout the deformation. Naturally, some base-points may become hidden because of a local intersection of the two bones of a joint being bent. We chose to keep these local intersections, since it will not degrade the resulting outer envelop of the model, which remains visually satisfying as if the surface became

flattened in the fold of a joint [VBG*13]. Furthermore, if the base-point associated to a detail-point is hidden, the detail-point is also hidden in the fold of a joint. Figure 3.1 illustrates that our heightfield skinning can also be used for meshes by skinning the vertices heights, while keeping the connectivity fixed, at the risk of creating self-intersections or slivers, a risk shared by many mesh skinning methods (e.g. [TGB13]).

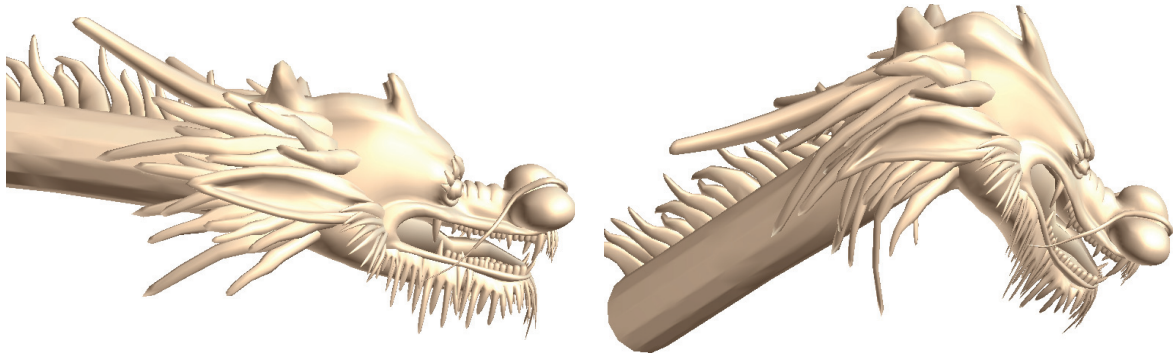


Figure 3.1 – Pose change for a mesh with folded surface sheets by heightfield skinning.

Algorithm 2 Pose change using heightfield skinning

Input: Sphere-mesh model registered to the input point set P .

Target pose and anatomy.

Output: Point set P' corresponding to the target changes.

- 1: Compute base point set \tilde{P} corresponding to P on the registered sphere-mesh and compute the set of height values $h(P)$
 - 2: Apply point set skinning on the base points \tilde{P}
 - 3: Deform \tilde{P} to the target pose and anatomy, yielding \tilde{P}_{sk}
 - 4: Re-project points of \tilde{P}_{sk} on the sphere-mesh yielding \tilde{P}'
 - 5: Lift points \tilde{P}' to the skin surface using the - possibly scaled - initial height of each point, yielding P' .
-

3.2 Base-point skinning

The skinning approach that we apply to the base-points is an alternative to classical skinning approaches [BP07, TGB13]. It could also have been used directly on the point-set details. Figure 3.2 compares the result of using a heightfield over base-point skinning with using directly point set skinning. Thus, in the following we describe the method in a generic point set context for notation simplicity.

The general idea of skinning is to attach each point to one or more bones with weights measuring the influence of each bone on it. The position of point after a deformation is a weighted linear

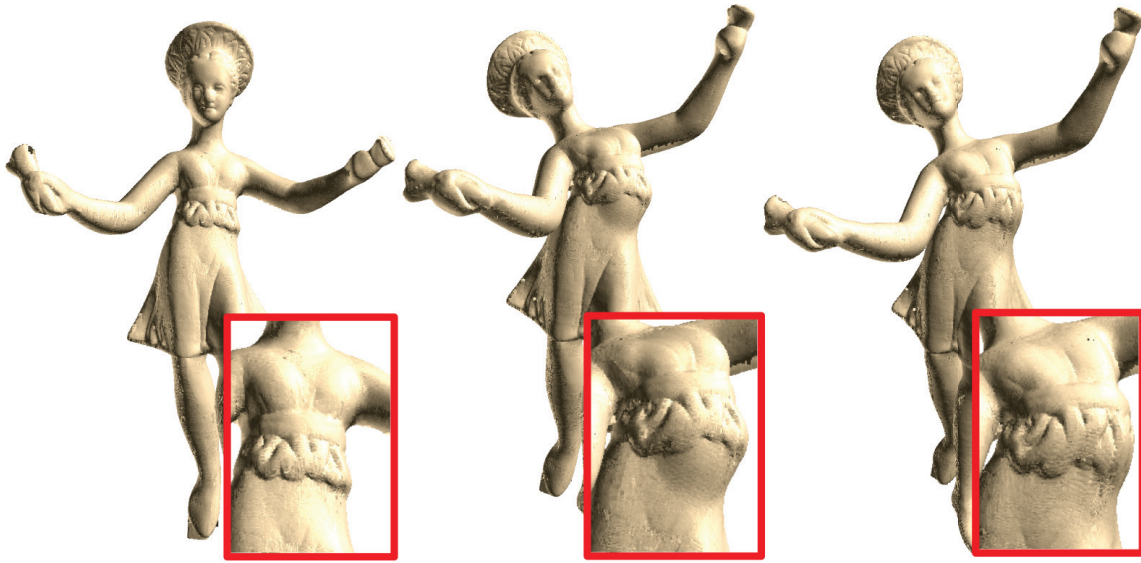


Figure 3.2 – Heightfield skinning. From left to right: original point set; pose change with direct point set skinning; pose change with heightfield skinning.

combination of the positions relative to its influencing bones. Linear Blend Skinning [MTLT88], one of the most popular skinning method, causes some well-known collapsing problems at joints, in particular for large rotations. For example, the volume at joint is not preserved when it is bent around its axis (Figure 3.5a). Similarly, if we apply a twist around the axis of a bone while keeping its predecessor fixed, Linear Blend Skinning produces a folding of the joint around a singular point (Figure 3.6a). These flaws are avoided by Dual Quaternions Skinning [KCvO07] which interprets a combination of rigid transformations as a rigid transformation, however artefacts still appear in the concavities (Figure 3.5).

We propose to deal with the pose change in a different way that breaks down the movement into its natural components. Similar as [KSH12], we consider that the motion between two bones at a joint is the combination of a twisting rotation around the axis of one of the bones and a bending rotation around an axis perpendicular to both bones' axes. From there, we introduce motion-dependent skinning weights: the weight of a point is not the same for twisting and bending rotations. The impact of bending on the base point set should naturally be limited to an area loosely enclosing the rotating sphere joint, while the impact of twisting obviously extends to the length of the bones adjacent to the twisted articulation. This is more coherent with the fact that underlying muscles are arranged along the bones and attached to the joints. In our approach, a point $p \in P_k$ can only be influenced by the bones adjacent to B_k so that it has at least one and at most three weights.

3.2.1 Bending rotation with anisotropic weights

In order to overcome the artefacts that arise in the convex part of joints with Linear Blend Skinning, we use Dual Quaternions Skinning [KCvO07]. The weights of a point p during a bending rotation follow a Gaussian profile of the distances from p to each of the bones that influence it. It is driven by a parameter ε_r controlling the size of the influence area (Figure 3.3). In our experiments, we automatically set ε_r larger than the average distance between the point set and our model. If the skinning process is used on base-points on a sphere-mesh, we set ε_r value for a joint as a function of its radius and the length of the adjacent bones :

$$\varepsilon_r = 0.2 * \min(r_{k+1}, l_k - r_{k+1} - r_k, 0.5l_{k+1} - r_{k+1} - r_{k+2}).$$

The weight $\omega_j(p)$, relative to one of p 's influencing bones B_j writes:

$$\omega_j(p) = \frac{1}{c} \exp -\frac{\|p\tilde{p}_j\|^2}{2(\varepsilon_r/3)^2} \quad (3.1)$$

where c is a normalizing factor ensuring that the weights sum to 1, and \tilde{p}_j is the orthogonal projection of p onto B_j . Hence if p is on bone B_j (as is the case for base-points), $\tilde{p}_j = p$. Figure 3.3 shows the weights profile along two bones.

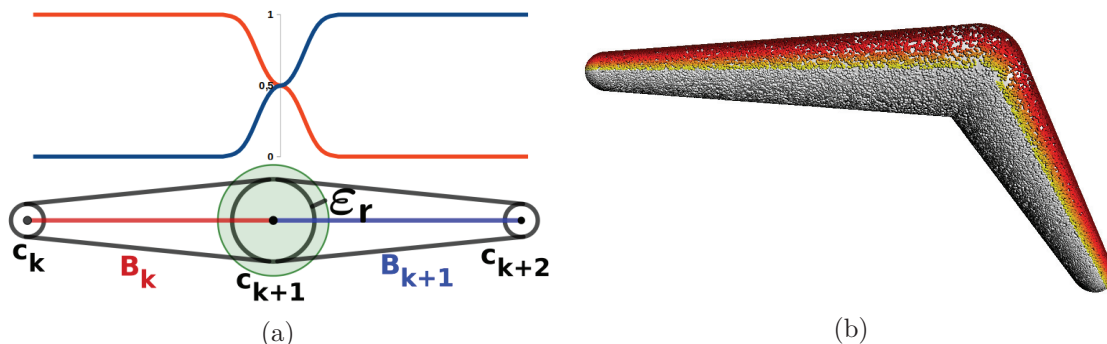


Figure 3.3 – Skinning weights for the bending rotation around a joint. (a) the red curve represents the influence weights of the left bone B_k and the blue curve represents the influence of the right bone B_{k+1} . The influence area of each bone is controlled by ε_r . (b): values of anisotropic ε_r on a bone. Points such that $\varepsilon_r = 0$ are shown in white. From yellow to red, the value of ε_r increases linearly.

However, it is possible to further improve the Dual Quaternion skinning method, by noticing that if ε_r is large, the convex part behaves well with points sliding from the conic parts of both bones to the joint sphere, while an artefact is created in the concave part. On the contrary, if ε_r is close to 0, which corresponds to no skinning at all, a hole appears in the convex part, but a self-intersection is created in the concave part (Figure 3.4). Quite counter-intuitively, this self-intersection is much more visually satisfying than the thinning artefact observed otherwise. Indeed, only the outer envelop is visible and, when allowing self-intersection, this envelop is

similar to the one obtained if a contact area was computed between the two bones [VBG*13]. To get the most of the two possibilities, we propose to adapt ε_r so that it is close to 0 for points in the concave part and it reaches its larger value for points on the convex parts. More precisely, ε_r varies continuously with respect to an *anisotropy angle* (See Figure 3.3).

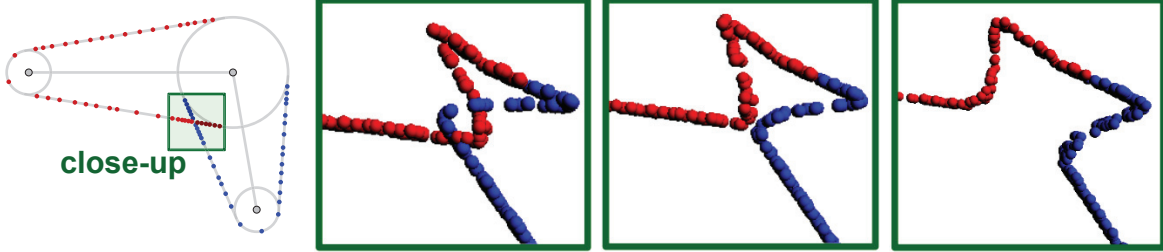


Figure 3.4 – Influence of ε_r on the skinning resulting from a bending rotation. In the first figure, $\varepsilon_r = 0$. The value ε_r increases from left to right.

We define the anisotropy angle α_b at p as the angle between the plane defined by B_k and B_{k+1} axes and the plane defined by B_k 's axis and p . This angle allows for a continuous transition over the skinned surface between points with no skinning, favoring local self-intersections ($\alpha_b \leq \pi/2$), and points with skinning, favoring the diffusion of points over the spherical joint surface ($\alpha_b \geq \pi/2$). Here, ε_r is deduced from the anisotropy angle as $\varepsilon_r = \cos \alpha_b \varepsilon'_r$, with ε'_r controlling the influence area size. Therefore, the weight $\omega_j(p)$ associated to point p and one of its influencing bone B_j is:

$$\omega_j(p) = \begin{cases} \frac{1}{c} \exp -\frac{\|p\tilde{p}_j\|^2}{2(\cos \alpha_b \frac{\varepsilon'_r}{3})^2} & \text{if } \cos \alpha_b < 0 \\ 1 & \text{if } \cos \alpha_b \geq 0 \text{ and } p \in P_j \\ 0 & \text{if } \cos \alpha_b \geq 0 \text{ and } p \notin P_j \end{cases} \quad (3.2)$$



(a) Linear blend skinning (b) Dual Quaternions skinning (c) Our anisotropic skinning

Figure 3.5 – Comparison of our anisotropic skinning method with Linear Blend Skinning and Dual Quaternions for a bending rotation. For a fair comparison, the two first methods use the Gaussian weight of Equation 3.1 which is made anisotropic in our anisotropic skinning. Dual Quaternions fix the volume collapse of Linear Blend Skinning near the convexity, but artefacts remain in the concave part, while our skinning method does not suffer from any of these flaws.

3.2.2 Twisting rotation

Let us consider a bone B_{k+1} twisted around its axis by an angle β , with its predecessor B_k kept fixed. Such a rotation impacts points attached to bones B_k and B_{k+1} . To handle this twist, we replace the linear combinations of bone motions with rotations adapted to the points. More precisely, each point is rotated around B_k 's (resp. B_{k+1}) axis by an angle that depends on its distance to B_k (resp. B_{k+1}). For $p \in P_k$ rotating around the axis of B_k , this angle writes $\beta(p) = \omega(p)\beta$ with

$$\omega(p) = \frac{\|v_{k+1} - p^*\|}{\|v_{k+1} - c_{k+1}\| + \|c_{k+1} - w_{k+1}\|} \quad (3.3)$$

p^* is the projection of p on B_k 's axis, and v_{k+1} (resp. w_{k+1}) is a point on B_k 's axis (resp. B_{k+1}) delimiting the impacted areas (Figure 3.7). By default $v_{k+1} = c_k$ and $w_{k+1} = c_{k+2}$, but different impacted areas can be designed by choosing different v_{k+1} and w_{k+1} . Here, the expression for $\omega(p)$ corresponds to a linear evolution of the rotation angle along the bone, but other types of influences could be designed. On Figures 3.6 and 3.7, we compare our method with Linear Blend Skinning and Dual Quaternions for a twisting rotation: the trace of the points initially aligned on a segment is much smoother with our approach.



Figure 3.6 – Twist motion: near blend Skinning and Dual-Quaternions Skinning to be compared with our approach in figure 3.7 that uses twist specific weights. Green dots show points that were aligned before the twisting rotation.

3.2.3 Combination of twist and bend.

Twist and bend specific skinning are combined to get the skinning result for any rotation around a joint. We perform first the twisting rotation and then the bending rotation. The positions and the weights of the points relative to each bone are updated between these two specific rotations. Figure 3.8 shows a comparison of our approach with Linear Blend Skinning and Dual Quaternions on synthetic data with a two-bones sphere-mesh model. As was expected, Linear Blend Skinning yields the well-known collapse at the joint. To alleviate this effect, bending with Dual Quaternions can be used, with a region of influence restricted to the neighborhood of the joint but some artefacts remain in concave area. However, the twist is handled differently in our approach and its effect is distributed over the length of the bone, yielding a more natural result.

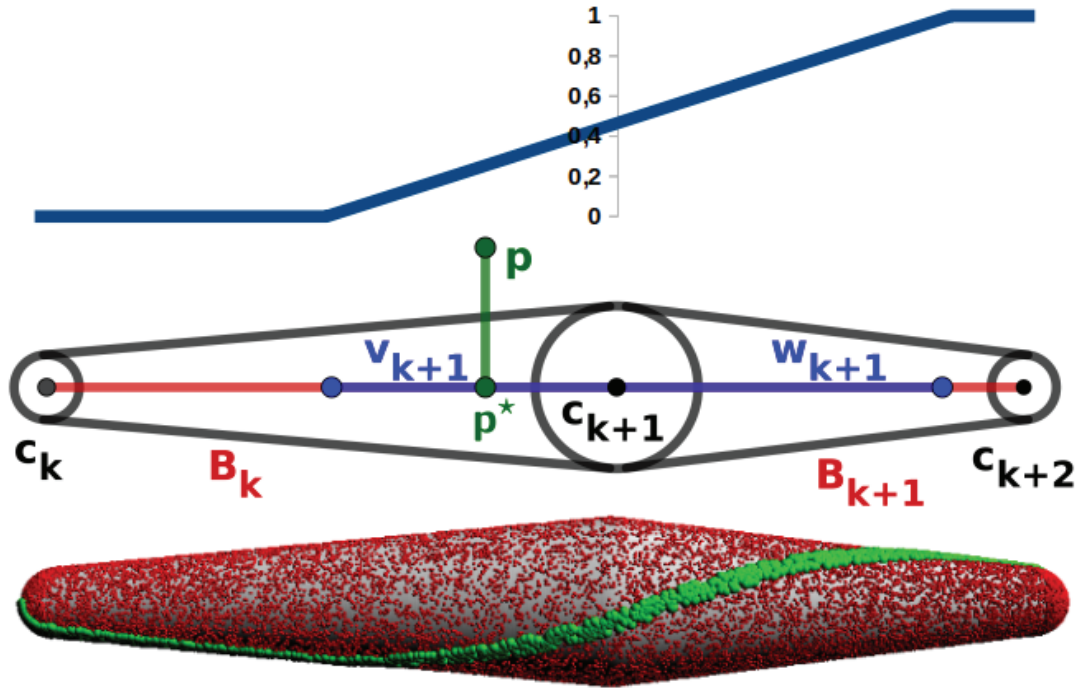


Figure 3.7 – The blue curve represents the value of ω_β . Points v_{k+1} and w_{k+1} define the range of the twisting effect of B_{k+1} . p^* is the projection of p on the skeleton line.



Figure 3.8 – Comparison of skinning approaches on a sphere-mesh model with combined twist and bend motions. Left to right: Isotropic Linear Blend Skinning, Isotropic Skinning with Dual Quaternions, Ours

3.3 Results

In this section, we show the performance of our skinning method and results of virtual restoration. We developed our algorithm in C++, using OpenMP for computing point to bone distances in parallel. All experiments are run on an Intel Core i7-4790K CPU @ 4.00GHz.

Figure 3.9 shows the FAKIR registrations on four statues and compares resulting models after skeleton pose change and skinning with our method or skinning with Dual Quaternions. Our skinning method clearly improves the quality near bone joints. As can be seen in particular in the areas circled in red in the fifth column, our method suppresses or at least reduces the Dual Quaternion artefacts around the bone joints. For 500000 points, our skinning approach takes about 1s, including computation of base points and re-projection.

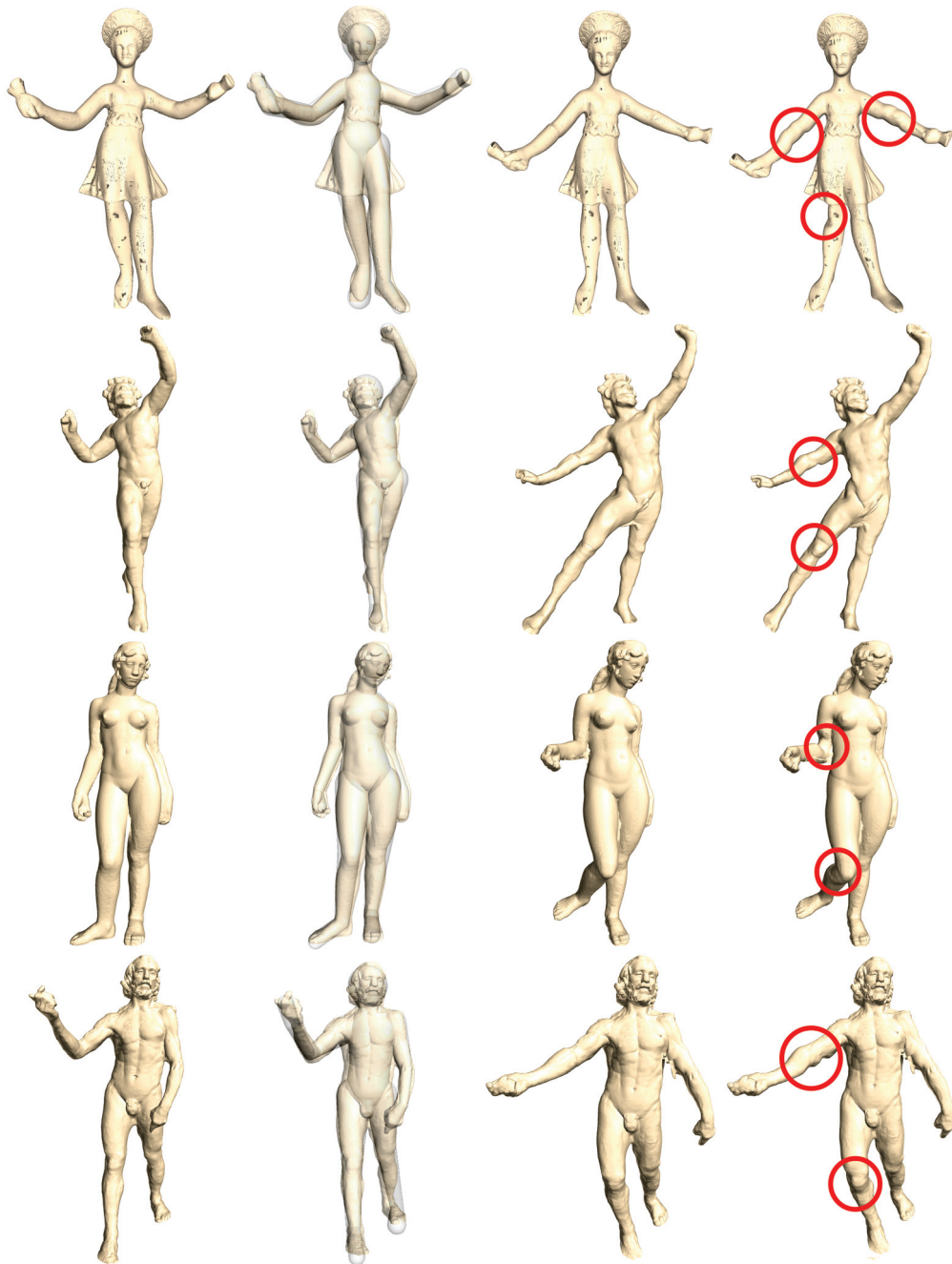


Figure 3.9 – Registration and pose changes of 4 statues: the Dancer with Crotales (first row), the Dancing Faun (second row), Aphrodite (third row) and the Old Man Walking (fourth row). First column: initial point set, second column: overlay of the registered model and the point cloud, third column: final point set in a modified pose by our skinning method, fourth column: skinning result with Dual Quaternion.

A SECOND APPROACH TO SPHERE-MESH MODEL SKINNING

Skinning techniques are widely used in the context of character animation. In the animation research community, *skinning* process allows deforming the skin following an underlying skeletal animation, which is a widespread method to animate 3D models. In most skinning methods, the skin is a 3D mesh whose vertices are attached to a set of bones using different weights, and the skeleton is a tree whose nodes represent joints of the skeleton and edges represent bones. Unlike the animation of 3D characters, skinning for virtual restoration is more sensitive to preserving the initial surface details. Besides, it demands not only rigid transformations but also affine transformations such as scaling. The existed geometric skinning methods, for example, linear blend skinning [MtLTM88] or dual quaternion skinning [KCZO08], suffer volume collapse problems and need to set appropriate weights. Example-based methods [MTG03] and physics-based methods improve artefacts of geometric skinning. However, they are computationally expensive, and both need additional input data. Our goal is to provide a skinning technique that gives a logical result and keeps the original precision for the point set of statues.

In this chapter, we introduce another skinning approach called Baseline skinning. Our baseline skinning method is based on a registered sphere-mesh model and works directly on a point set. The registered model is built from the FAKIR algorithm introduced in chapter 2. We propose considering the skin of the model as a set of detail points, encoded as a set of displacements on top of our elementary articulated model and transferred back on the model after pose or intrinsic parameters changes. We associate each detail points to a base-point on the articulated model so that each point has a relative position concerning the model. The principle of our baseline skinning is that a baseline drives each base-point on the model, which moves with the model. After applying skinning on base-points, we lift base-points by adding relative displacement values to recover the point set after deformation. The baseline movement mimics muscle's change, and base-points may locally slide on the moving baseline to mimic the sliding of the skin on joints. Meanwhile, our point set skinning process does not need to compute weights. In the following, we explain the baseline and base-point definition for a giving point and give the algorithm of baseline skinning.

4.1 Baseline

In our articulated model, we consider each sphere-mesh chain of bones as a union of baseline curves composed of a succession of segments and circle arcs. Each circle arc involved in the baseline curve is coplanar with the segment that precedes it and the segment that follows it (See green curve Figure 4.1). A circle arc does not separate two consecutive segments if their supporting generator lines intersect on the bones' conic part (See blue curve in Figure 4.1).

In the following, we take an example of one point p and a pair of bones to show the baseline of point p on the model. We first project p on its closest bone B_1 . \tilde{p} is the orthogonal projection of p on the bone B_1 and p_b is the base-point of p (See in Figure 4.1). Let consider one portion of the chain that is composed of a sphere joint (c_2, r_2) and its two tangent pieces of cones on B_1 and B_2 , which are tangent to the spheres (c_1, r_1) and (c_2, r_2) respectively. Let A_1 and A_2 denote the apices of the two cones. By intersecting the sphere-mesh portion with a plane including the line A_1A_2 and point \tilde{p} , one obtains pieces of the baseline curves for point p . Each piece of the curve comprises two segments joined by a circle arc when they do not share a common extremity. More precisely, there are always two intersection paths on the model for one point except when plane $A_1A_2\tilde{p}$ is the tangent plane to two bones. One intersection path is showed in green in Figure 4.1 which contains two segments (UV and XY in green in Figure 4.1b) and an arc (\widehat{UX} in green in Figure 4.1b and point S is the midpoint of the arc). The other is showed in blue which has two segments (US and SY in blue in Figure 4.1b and the point S is the intersection of two segments UV and XY). We choose the one that \tilde{p} is on it. So in Figure 4.1b the blue one is the real baseline for point p . Points U , V , X , Y and S are key points of the baseline.

Indeed, such a plane intersects the cones through generator lines and the joint sphere through a circle. By intersecting the portion mentioned above of the sphere-mesh for varying positions of the plane containing the line A_1A_2 , one can decompose the shape portion as a union of curves. If one of the cones degenerates into a cylinder, the varying plane should include the line that goes through the remaining cone's apex, parallel to the cylinder axis. If both cones degenerate into cylinders, the plane intersecting the portion should be parallel to the cylinders' axes.

4.1.1 Baseline before transformation

In our method, each point is attached to its closest bone. The position of one point for a movement is influenced by its closest bone, the predecessor bone and the successor's bone. So the baseline of one point passes two or three bones. In our skinning process, we define for each point its initial baseline before transformation and its baseline after transformation.

Generally, for a point p_k attached to bone B_k , we need to find the initial baseline passes through its predecessor bone B_{k-1} , its attached bone B_k and its successor bone B_{k+1} . To do so, we consider each time a pair of bones. Taking example in Figure 4.2a, for B_0 and B_1 , the

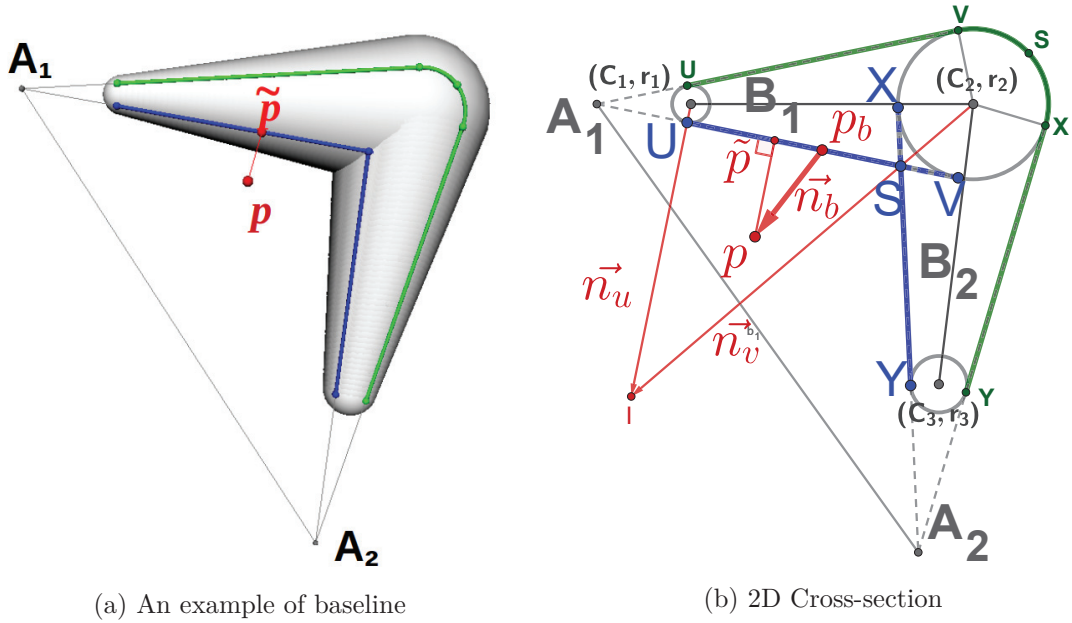


Figure 4.1 – Illustration of baseline and base-point

intersection path consists of segments MS_1 and S_1V . We get key points M, N, U, V and S_1 . They are all in the plane $A_0A_1p_b$. Similarly, we can get key points U, V, X, Y and S_2 for B_1 and B_2 , which are all in the plane $A_1A_2p_b$. Segment UV is on the intersection line of plane $A_0A_1p_b$ and plane $A_1A_2P_B$. In this example, the baseline of point p composes by segment MS_1 , segment S_1S_2 and segment S_2Y . A complete baseline contains 8 key points M, N, U, V, X, Y, S_1 and S_2 .

4.1.2 Baseline after transformation

when bones move, the base-point p_b of point p and its baseline move also with bones. Let $trans_k(\cdot)$ represent the function of the transformation for $Bone_k$. We take the example of two bones like in Figure 4.1b and assume that bones transform in a new position which is shown in Figure 4.3a. The new positions for key points of the baseline are $trans_0(U)$, $trans_0(V)$, $trans_1(X)$ and $trans_1(Y)$. To simplify we use still U, V, X and Y to represent new positions of key points. So segment UV and segment XY are no longer coplanar. We illustrate this in blue segments in Figure 4.3a and Figure 4.3b respectively for the case of two bones and three bones). For this reason, we can not use directly segment UV and segment XY to define the baseline after a transformation. Instead, we take a baseline in middle of segment UV and segment XY which we show in Figure 4.3a in yellow: segment $U'S'$ and $S'Y'$. S' is the intersection of segment $U'V'$ and segment $X'Y'$. Point V' and point X' are not shown in the Figure 4.3a. More details is in section 4.4. In general, we get key points M', N', U'_1, V'_1 and S'_1 for bones B_{k-1} and B_k ,

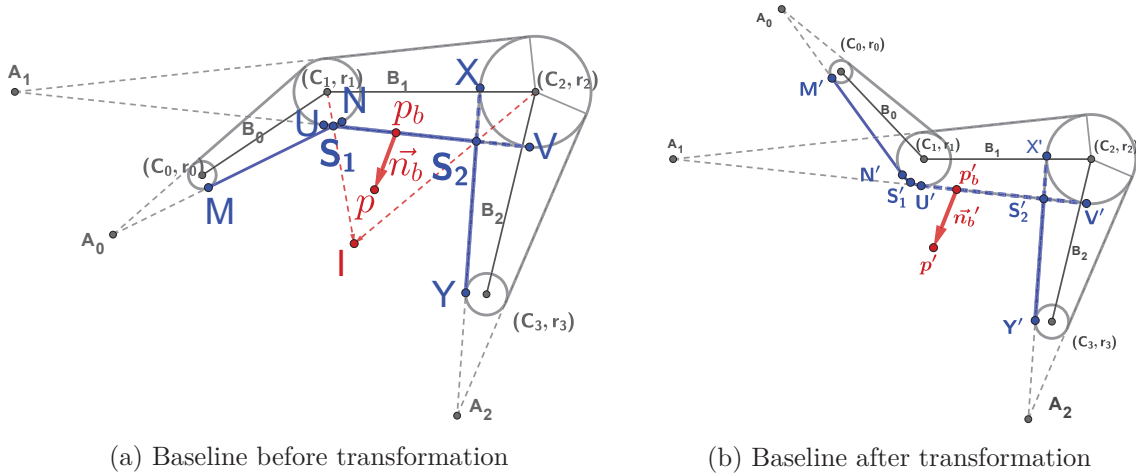


Figure 4.2 – Illustration of baseline before and after a movement

U'_2, V'_2, X', Y' and S'_2 for bones B_k and B_{k+1} . As showing in Figure 4.3b, the red segment $S'_1 V'_1$ and the yellow segment $U'_2 S'_2$ are two different segments. So we need an interpolation process between S'_1 (U'_1 if there is an arc between two segments) and s'_2 (V'_2 if there is an arc between two segments) to keep the continuity on the baseline curve when we determine the baseline after a transformation. We introduce our interpolation scheme in section 4.2.1.

Finally, the baseline after transformation for one point is defined by key points: $M', N', U'_1, S'_1, V'_2, S'_2, X'$ and Y' (See in Figure 4.3b).

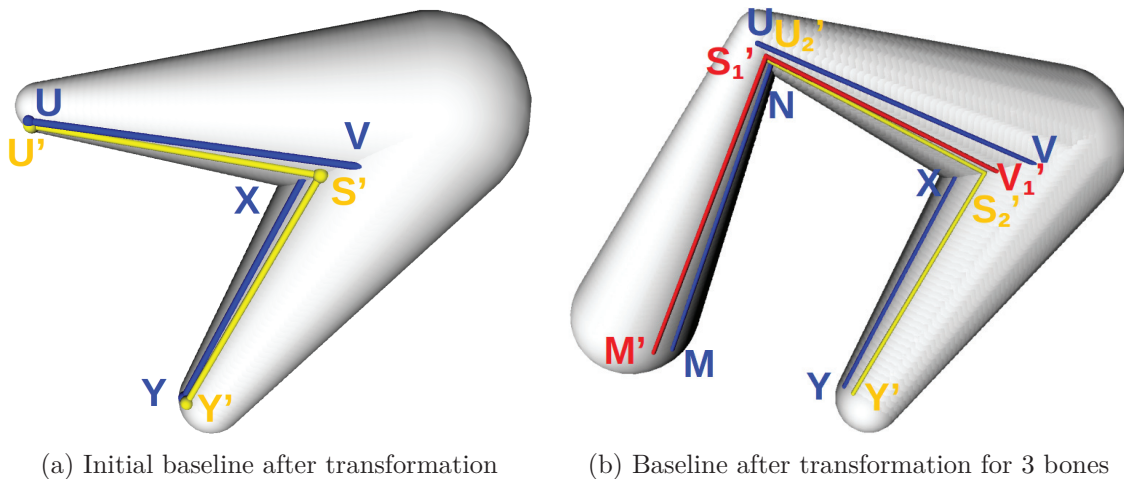


Figure 4.3 – Illustration of baseline and base-point after transformation

4.2 Base-point

Each base-point p_b of a baseline curve is associated with a direction of detail \vec{n}_b . We do not use directly \tilde{p} as the base-point of point p . Instead, the base-point p_b depends on the direction of detail \vec{n}_b and \tilde{p} helps to fix p_b and \vec{n}_b . If the point is on a circular arc of the sphere (c_2, r_2) , the direction of detail is given by the unit vector pointing from c_2 towards \tilde{p} . In this case, \tilde{p} is the base-point of p . If \tilde{p} is on a segment part supported by B_1 (resp. B_2), its direction of detail is enclosed in the plane including B_1 's axis (resp. B_2 's axis) and \tilde{p} . Therefore, directions of detail are coplanar along a baseline segment but not along a circular arc and not from one segment of the curve to another (See figure 4.8).

Along a segment, directions of details are distributed continuously with some constraints at the segment extremities and the coplanarity constraint. At one extremity, one should pass continuously from the direction of detail on the segment to the direction of detail on an incident arc of a circle or an incident segment. If a segment is adjacent to two circle arcs on the baseline curve, the direction of detail locally corresponds to the cone's normal. Though, if a segment on B_k is incident to another segment on B_{k+1} , the direction of detail at their common extremity S is given by the unit vector pointing from the center c_{k+1} of the sphere towards S . We show this case in Figure 4.1b. The direction of detail \vec{n}_b is set by an interpolation between $\vec{c}_1\vec{U}$ and $\vec{c}_2\vec{S}$. c_1 and c_2 are centers of spheres. In the following, we note $\vec{c}_1\vec{U}$ as \vec{n}_u and $\vec{c}_2\vec{S}$ as \vec{n}_v . The interpolation can be set as linear, quadratic, cubic, etc. If we use a linear interpolation to define \vec{n}_b :

$$\vec{n}_b = (1 - t) * \vec{n}_u + t * \vec{n}_v$$

where t is the index of the base-point position: The base-point of p is defined by:

$$p = p_b + h * \vec{n}_b$$

where h is the displacement value for point p .

Figure 4.8 illustrates a linear distribution of the directions of details over each segment of a baseline curve, but any other continuous distribution could be chosen. That way, one or several detail points can be given above a baseline curve point in the associated direction of detail.

4.2.1 Base-point after transformation

As we discussed in the section 4.1.2, the key points S'_1 and S'_2 (or U'_1 and V'_2 if there are arcs between segments) are not in the same plane after a pose and anatomy change. To keep the continuity of the baseline

Interpolation for base-point position

We apply an interpolation from U'_1 to V'_2 when we determine the position of base-point p'_b . We note respectively plane $c_1c_2U'_1$ and $c_1c_2V'_2$ as plane u and plane v in the following. To do so, we rotate the plane u to the plane v to find an interpolation between point U'_1 and V'_2 . The angle between plane u and plane v is α . When rotating plane u to plane v $\alpha_p(t)$ degrees, we get intersection points U^* and V^* between the rotating plane and the bone. To compute point U^* and point V^* , we can consider point U'_1 rotates $\alpha(t)$ degrees around $\overrightarrow{c_1c_2}$ and point V'_2 rotates $1 - \alpha(t)$ degrees around $\overrightarrow{c_2c_1}$. So the positions of points U' and V' is computed by:

$$U^* = \text{Rotation}(\overrightarrow{c_1c_2}, \alpha_p(t))(U'_1)$$

$$V^* = \text{Rotation}(\overrightarrow{c_2c_1}, (1 - \alpha_p(t)))(V'_2)$$

The position of base-point p'_b after transformation is:

$$p'_b = (1 - t) * U^* + t * V^*$$

Where $\alpha_p(t)$ is the function of the interpolation strategy. It is controlled by t and decides the rotation angle for interpolation between point U'_1 and V'_2 . This interpolation can be linear, quadratic, cubic, or users can define it according to the desired result. We give here a linear and a cubic interpolation functions which we have used in our experiences:

$$\alpha_p(t) = t\alpha$$

$$\alpha_p(t) = -2 * \alpha * t^3 + 3 * \alpha * t^2$$

Interpolation for detail direction of base-point

The definition of $\alpha_d(t)$ is similar as $\alpha_p(t)$. We can use linear interpolation, quadratic interpolation, cubic interpolation or other type of the interpolation that a user can choose. If $\alpha_d(t) = \alpha_p(t)$, the detail direction $\overrightarrow{n'_b}$ of one point p'_b is in the same plan which consists of the axe of the bone and point p'_b . But it is not necessary that $\alpha_p(t)$ and $\alpha_d(t)$ are identical. $\overrightarrow{n_u}$ and $\overrightarrow{n_v}$ are the direction of details on point S'_1 (or U'_1) and S'_2 (V'_2). $\overrightarrow{n_u}$ is in the direction $\overrightarrow{c_1S'_1}$ (or $c_1U'_1$) and $\overrightarrow{n_v}$ is in the direction $\overrightarrow{c_2S'_2}$ (or $c_2V'_2$). We illustrates in Figure 4.2b. After an interpolation:

$$\overrightarrow{n'_u} = \text{Rotation}(\overrightarrow{c_1c_2}, \alpha_d(t))(\overrightarrow{n_u})$$

$$\overrightarrow{n'_v} = \text{Rotation}(\overrightarrow{c_2c_1}, (1 - \alpha_d(t)))(\overrightarrow{n_v})$$

The direction of detail of point p'_b is defined by:

$$\vec{n}_b' = (1 - t) * \vec{n}_u + t * \vec{n}_v$$

4.3 Baseline skinning process

We show an entire baseline for point p before and after transformation in Figure 4.2. In the initial position, we define baseline b with key points, base-point p_b and distance h for point p as well as t to indicate the relative position of p_b on the baseline. t is computed by:

$$t = \frac{\text{length}(S_1, p_b)}{\text{length}(S_1, S_2)}$$

Where $\text{length}(S_1, p_b)$ and $\text{length}(S_1, S_2)$ are not always distance between two points but are sometimes a combination of the length of a segment and the length of an arc or two arcs. The value of h equals to the distance between point p and its base-point p_b :

$$h = \|pp_b\|$$

While deforming the model, the baseline b moves with the model so that we get the new position of the baseline b' which the key points of b' are: $M', N', U', V', X', Y', S'_1$ and S'_2 (See in Figure4.2b). Then we use t to determine the position of p'_b on the new baseline b' after the transformation of the model. The position p'_b meets the condition:

$$\text{length}(S'_1, p'_b) = t * \text{length}(S'_1, S'_2)$$

Finally, the new position of point p after a movement is computed by:

$$p' = p'_b + h * \vec{n}_b'$$

Algorithm 3 Pose and anatomy change using baseline skinning

Input: Sphere-mesh model registered to the input point set P .

Target pose and anatomy.

Output: Point set P' corresponding to the target changes.

- 1: Define for each point $p \in P$ its base-point p_b , baseline b on the registered sphere-mesh and compute h and t
 - 2: Define the baseline b' of point p on the target pose and anatomy
 - 3: Determine p'_b and \vec{n}_b' by using t on baseline b' for target position
 - 4: Lift point p_b to the skin surface using h and \vec{n}_b'
-

4.4 Geometry

4.4.1 The intersection of two cones

Our process for defining the baseline curve considers each time a pair of bones. This model can be regarded as a combination of two intersected cones and a sphere which is tangent to two cones. There are different types of intersection for two cones. Figure 4.4 shows the type of intersection in our case: two cones have two common tangent planes but different vertices and do not have a common generatrix. Their intersection curve degenerates into two conics.

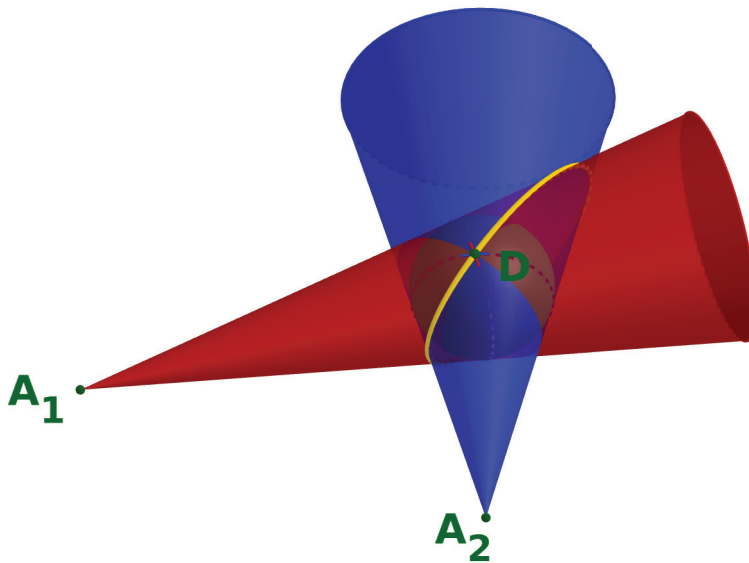
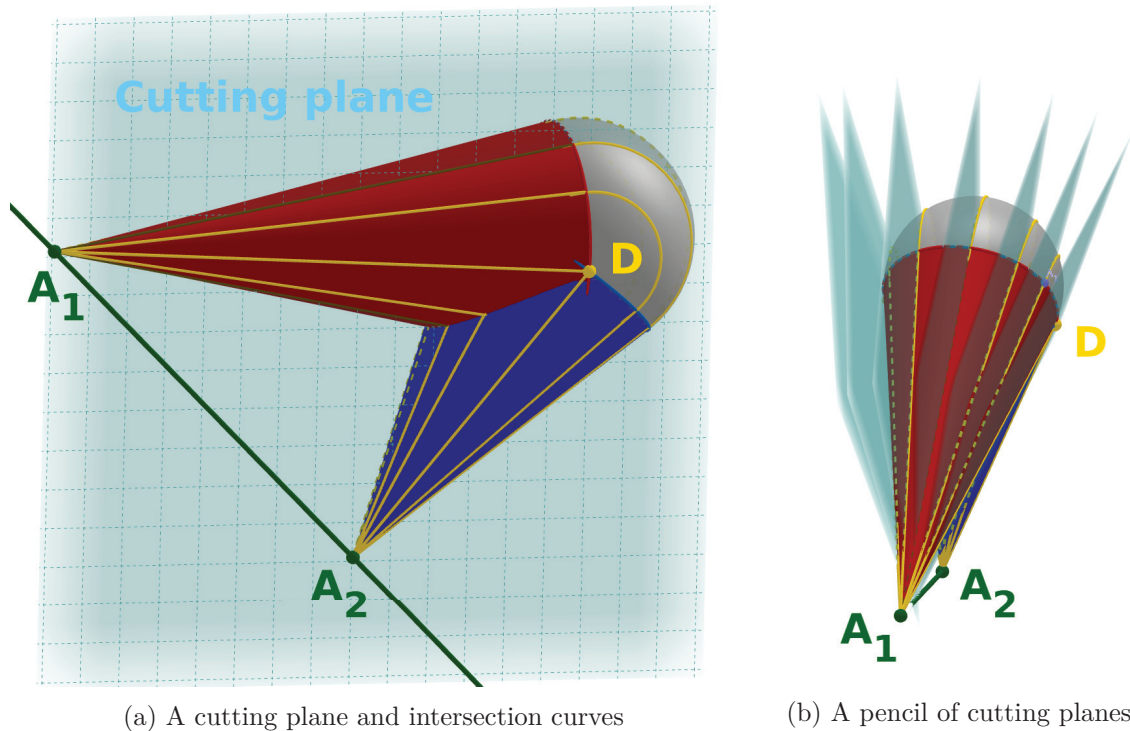


Figure 4.4 – Intersection of two cones

A general principle for constructing points of the intersection of two surfaces uses the method of cutting planes. To construct the intersection curves of our model, we choose a system of cutting planes that intersect both cones along their generatrices, a pencil of planes containing the line through both apices of the cones. Such a cutting plane can intersect a cone in two real and different generatrices, in one generatrix when a plane is a tangent plane. If a cutting plane intersects both cones in one real generatrix, this plane is a common tangent plane, and the intersection of these two generatrices is a double point of the intersection curve. As is shown in Figure 4.4, point D is a double point of the intersection curves and point A_1 , A_2 are apices of two cones. If the cutting plane passes the sphere surface, then the intersection curves consist of two generatrices and an arc. Figure 4.5a shows a cutting plane and several intersection curves on our model. Figure 4.5b shows a pencil of cutting planes for our model.



(a) A cutting plane and intersection curves

(b) A pencil of cutting planes

Figure 4.5 – Cutting planes and intersection curves on our model

4.4.2 The cutting plane after transformation

In the model's initial position, the cutting plane for one point p is constructed by its orthogonal projection \tilde{p} on the surface and two cones' apices. However, we can not use the same way to define the cutting plane after transforming the model. The reason is that the new position of point p is unknown since the model is changed. We need another point except two apices to construct the new cutting plane.

We take the same notation as in Figure 4.3a. Figure 4.6a illustrates a baseline curve containing segments UV , XY and an arc \widehat{VX} . Figure 4.6b shows new positions of segment UV and segment XY after a movement of model. UV and XY are not in the same cutting plane.

So our goal is to look for a cutting plane that is positioned "at best" between the two plane baselines passing through V and X , respectively. Considering the bisector plane of plane A_1A_2V and plane A_1A_2X does not solve the problem. In figure 4.7, the yellow curve containing point V and point X_1 is the intersection curve of plane A_1A_2V ; and the blue curve containing point X and point V_1 is the intersection curve of plane A_1A_2X . We, therefore, come back to looking for a plane that would pass close to V_m and X_m (see in Figure 4.7b). V_m is positioned halfway between the two cutting plane (or between point V and point V_1 on arc $\widehat{VV_1}$) on the first cone and X_m on the second cone). This time we take the bisector plane of plane $A_1A_2V_m$ and plane $A_1A_2X_m$ as

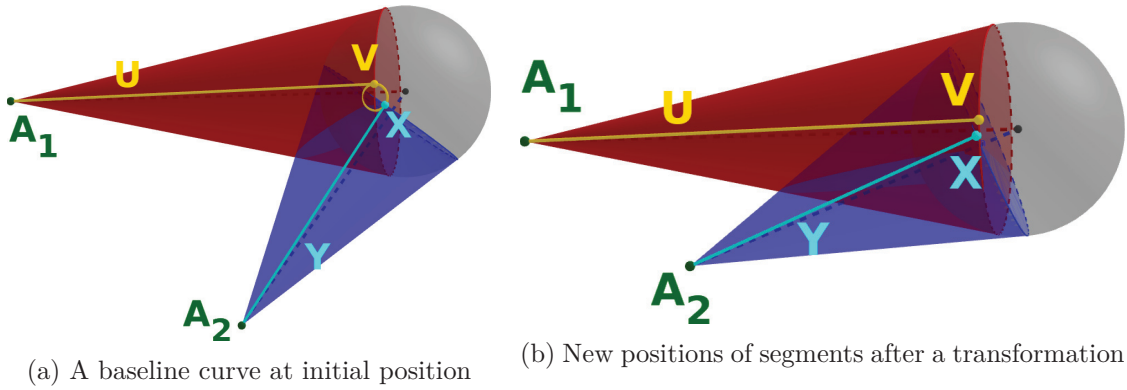


Figure 4.6 – The positions of segments on a baseline before and after a movement.

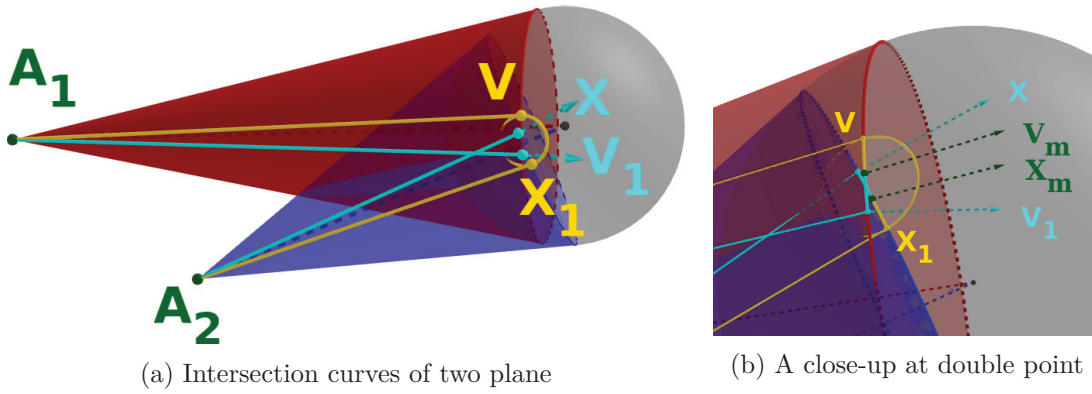


Figure 4.7 – The positions of segments on a baseline before and after a movement.

the cutting plane. This plane can give a better baseline in the middle position passing through V and X than the bisector plane of A_1A_2V and A_1A_2X .

4.5 Results

This section shows the performance of our baseline skinning method on synthetic data and real statue point set. We developed our algorithm in C++, using OpenMP for computing point to bone distances in parallel. All experiments are run on an Intel Core i7-4790K CPU @ 4.00GHz.

We first tested our algorithm on synthetic data. We generated a set of points lying on the same baseline and used a chain of three bones model. The attachment of points to bones is shown in different colors: red, green and blue. In Figure4.8, we show original points, base-points and their detail directions. The first row corresponds to a folding movement, and the second corresponds to an unfolding movement. The first column shows the initial position of point sets and models. The second column is our baseline skinning results by linear interpolation for both α_p and α_d . The third column is resulted by cubic interpolation. Our baseline skinning method

performs well in both situations. The skinning result can be adjusted by changing interpolation schemes. Since we do not use the orthogonal normal of a surface as the detail direction, we can avoid the intersection effect in the concave part between two bones.

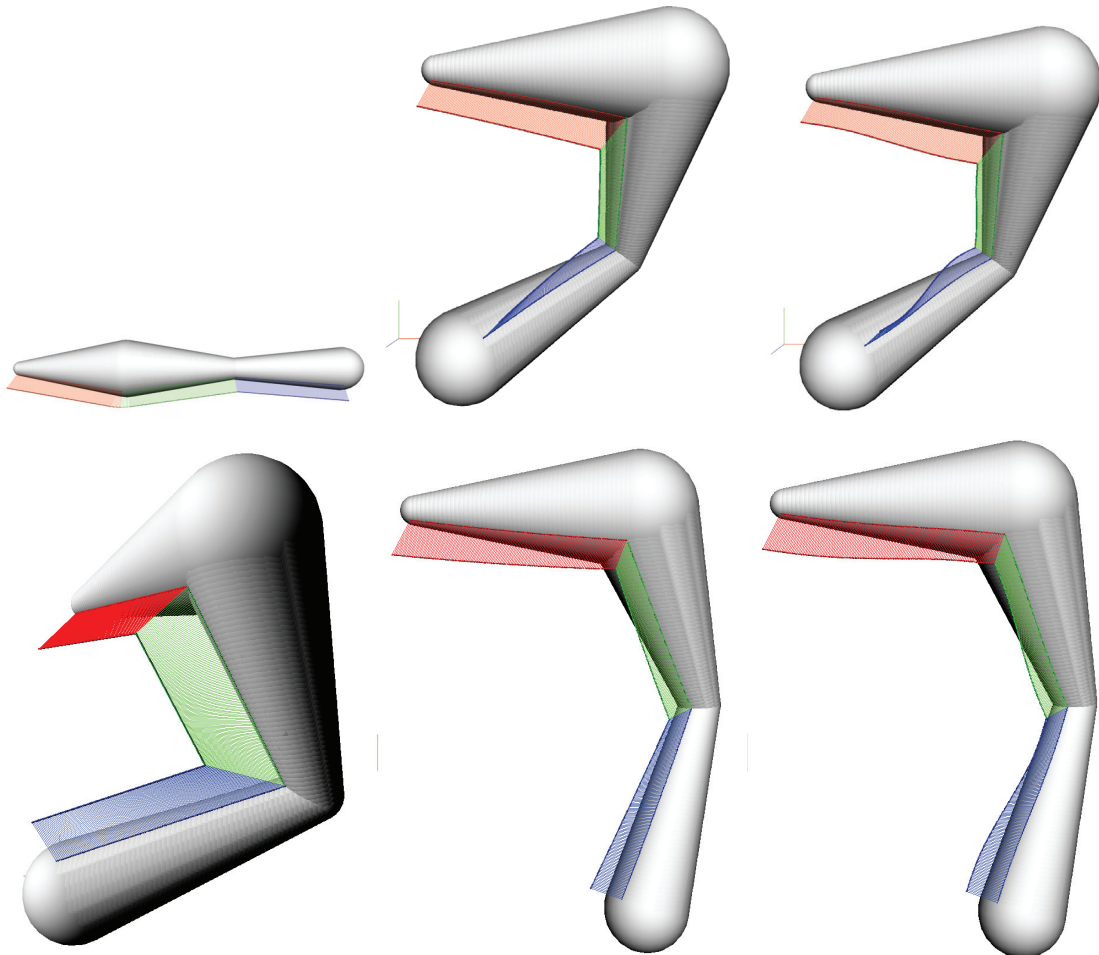


Figure 4.8 – Baseline skinning on synthetic data for a folding movement (first row) and an unfolding movement (second row). From left to right: initial position, linear interpolation for α_p and α_d , cubic interpolation for α_p and α_d .

We selected a set of statues from various sources:

1. Dancer with Crotales, Louvre Museum
2. Dancing Faun, Pompei excavations
3. Aphrodite, Thorvaldsens Museum
4. Saint John the Baptist, Ny Carlsberg Glyptotek

While the 'Dancer with crotales' is a raw point set. The other models are point sets sampled on meshes extracted from the Sketchfab website. Compared with dual-quaternion skinning, our

baseline skinning method improves the collapse or bulge effect at a joint and gives a more realistic result for the point set. However, a boundary line can be observed at the joint of two bones when unfolding bones, such as the right arm of the dancer at the third row in Figure 4.9. Another limitation is that our skinning method can only deal with a chain of bones. For this reason, we cannot deal with points that are influenced by two different chains. For example, the deformation of the dancer’s dress (third row in Figure4.9) can be influenced by the movement of two legs.

4.6 Conclusion and perspectives

We introduced a novel baseline skinning method for the point set. Our method does not require a weight computation for each point and gives good skinning results when changing a point set’s pose and anatomy. As future work, we would like to add a smooth process to polish the joint’s boundary line. Furthermore, we can associate different chains for dealing with points influenced by three or more bones.

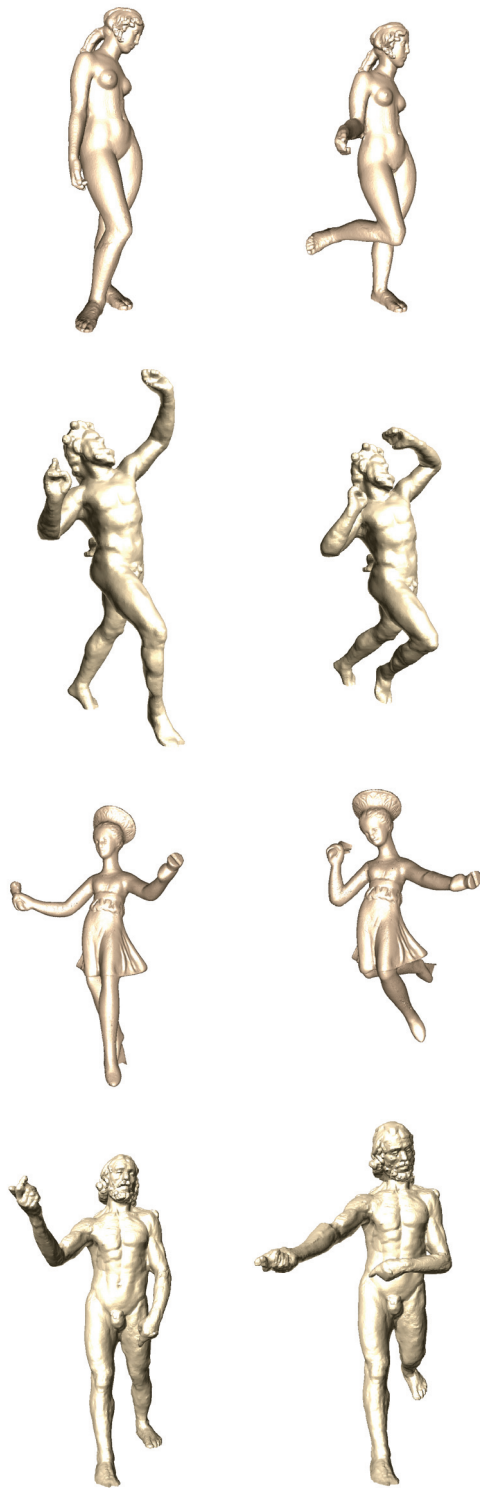


Figure 4.9 – Skinning results

VIRTUAL RESTORATION RESULTS

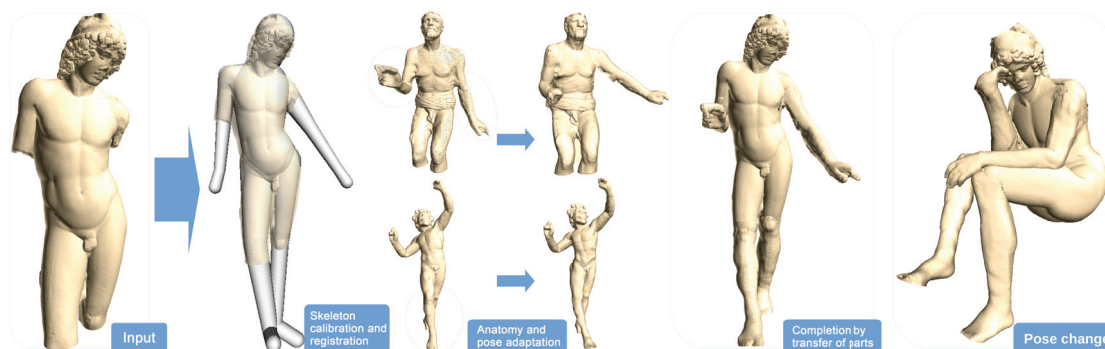


Figure 5.1 – Restoration and pose change of the Prince Paris statue. Through articulated model regression, we identify the anatomy and pose. We then adapt parts from other statues after a change of morphology and use them to complete the input. Once the statue is complete, we can also change its pose: here we bring it to the pose of *The Thinker* by Auguste Rodin.

Grossman et al. [GPT03] discuss varying approaches to the restoration of ancient sculptures. In particular, they discuss restoration through the combination of ancient fragments, a technique widely used in the 18th and 19th centuries.

Our goal is to replace missing parts of incomplete statues by deforming a point set of another statue. Our virtual restoration process contains three stages:

- Stage 1: Pose and anatomy estimation of statues. In this stage, we use FAKIR algorithm to register a sphere-mesh model to the target statue (the statue to be completed) and a sphere-mesh model to each of the statues used as candidates to complete the missing parts of the target statue.
- Stage 2: Pose and anatomy changes. We use a dedicated skinning method to change the dimension and position of candidate statues by using the same dimension and pose as in the target statue.
- Stage 3: Combination. We complete missing parts of the target statue by combining the corresponding parts of candidate statues.

In chapter 2, we introduced FAKIR algorithm which is used in the first stage. And we introduced two skinning methods in chapter 3 and chapter 4 that is used in the second stage. The last step

of our virtual restoration process is to combine parts belonging to different statues after bringing them to a common pose and anatomy. As a nod, we also illustrate that the technique of changing the pose of a statue can be used to generate new statues that could be used in the context of an animated film accompanying the description of a work in a museum.

In the following, we note the registered model of the incomplete statue as $M_t = \{B_k(l_t^k, \mathbf{r}_t^k), \Theta_t^k\}_{k=1, \dots, 22}$ and the registered model of the complementary statue, that is chosen to combine with the incomplete statue, as $M_c = \{B_k(l_c^k, \mathbf{r}_c^k), \Theta_c^k\}_{k=1, \dots, 22}$. The size of a limb of missing parts is corrected by its existing symmetric limb after registration, if it exists. For example, we change the size of left arm of Wounded Amazon (Figure 5.4, fourth row) by the intrinsic parameters of its right arm. In case there is no clue for the size of a limb, such as for the missing arm, forearm and hand of the Esquiline Venus (Figure 5.4, first row), we use default human proportions. We change the overall elementary anatomy and pose of complementary model to match the incomplete statue using our heightfield skinning process. The pose for the missing parts is the same as the one in the complementary statue, unless some clue is given by some partial data. The position of one point p is computed by:

$$p = \text{Skinning}(\text{Scale}(\frac{l_t}{l_c}, \frac{\mathbf{r}_c}{\mathbf{r}_t}, \frac{\mathbf{r}_c}{\mathbf{r}_t})p)$$

where *Scale* is a scaling function to make the anatomy of the complementary statue correspond to the incomplete statue and *Skinning* is our heightfield skinning function. Re-scaling the complementary statue to adapt the incomplete statue is necessary for the reason that the morphology of two statues can be quite dissimilar. We show in Figure 5.2 a comparison of restoration results with and without morphology change. The leg we use to complete statue Spinario is too long and too large (See right image in Figure 5.2). After scaling the length and the radii with the parameters of the registered model, the restoration result is more satisfying.

At last, we integrate the selected parts into the incomplete statue. We assume that the selected parts and the statue to complete slightly overlap, which is necessary to blend harmoniously statue parts. In this overlap area, information must be merged.

Provided the details are not multi-layered above each sphere-mesh model in the overlap area, the part merging consists in removing the lower layer, creating a sharp boundary between the point sets, and then blending the points near the boundary. Recall that our sphere-mesh model is used as a basis surface to express the residual heightfield information h after the registration. We propose to use it to combine the data points. Let p be a point and \tilde{p} its projection on the sphere-mesh model, we call Q_p be the subset of points of the two models that project on \tilde{p} , up to precision 3δ . The first step is to keep only the upper layer in the overlap area. To do so, we consider the subset H_p of the points in Q_p whose height values are larger than $h_{max} - \rho$, where h_{max} is the maximum height value for points in Q_p , and ρ is the average distance to the nearest neighbor for the points in the overlap area. Then we replace the height value of p by a

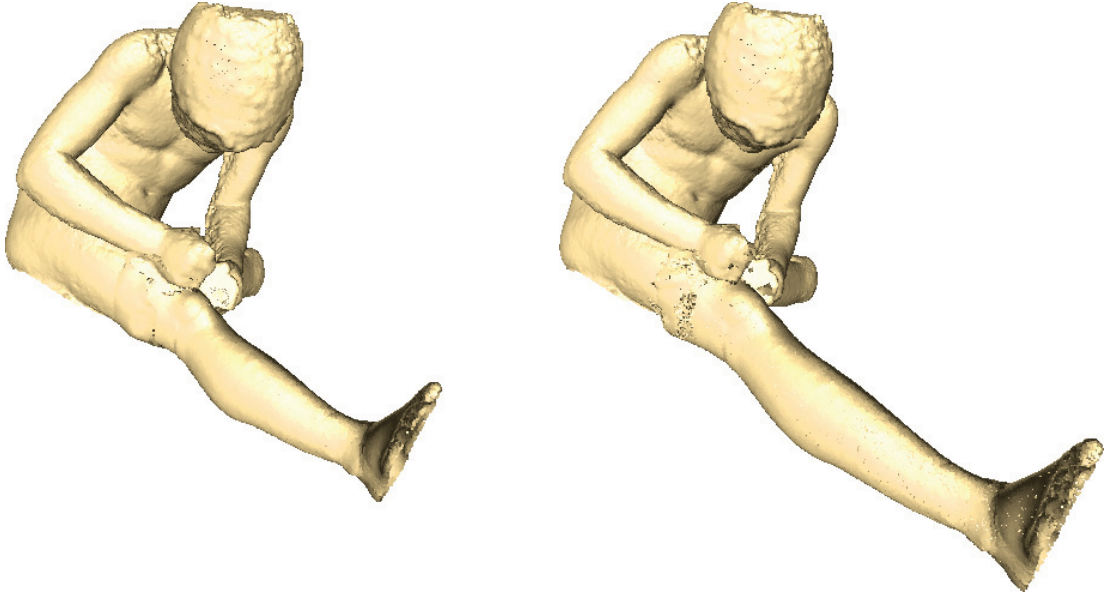


Figure 5.2 – The importance of morphology change in restoration. Left: restoration with scaling morphology of the leg part. Right: restoration using directly the original leg part.

Gaussian-weighted average of the height values in H_p . The resulting height value of p is thus:

$$h(p) = \frac{1}{S} \sum_{q \in H_p} e^{-\frac{\|p-q\|^2}{2(3\delta)^2}} h(q) \quad (5.1)$$

with S a weight normalizing factor. This brings the points of the lower layer in the overlap area to the upper layer, creating a sharp boundary. Finally, the sharp boundary is smoothed by Gaussian-weighted averaging of the heightfield values across the boundary. The default value of *delta* is set to $3\delta = 10\rho$. A high value of *delta* smoothes the merging area but can cause distortions.

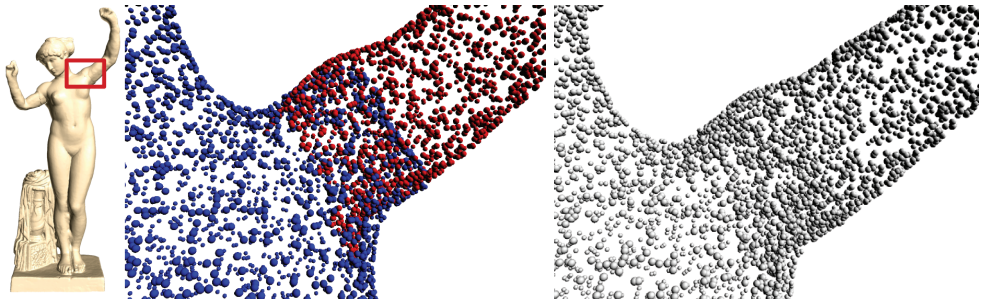


Figure 5.3 – Overlap area. Left: blue points and red points come from different statues. Right: Merging result.

In Figure 5.1, we show the process of completing a statue which lacks arms and legs. We can also change the pose of a statue after completion. For instance, we bring the Prince Paris statue to the pose of The Thinker statue. Indeed, sometimes the pose of a broken statue can not be deduced by its remain parts. In this case, we can give several hypotheses of its pose through our skinning algorithm.

Figure 5.4 shows the restoration of five other statues. The Esquiline Venus (first row) and the Venus de Milo (third row) are both missing arms while The Old Fisherman is missing legs (second row). The Wounded Amazon (fourth row) is missing an arm and the Spinario (fifth row) is missing a leg. The restoration method recovers plausible arms and legs in all five cases, leading to plausible restored statues. A limitation of our method is that sharp junction lines can be observed sometimes in the multi-layered area (See the Wounded Amazon in Figure 5.4). This due to the choice of δ in equation 5.1. We make a compromise between a smooth fusion and a preservation of styles of both parts. The arms we have chosen for the two Venus statues are from masculine statues. Although the morphologies are quite different, our restoration merges the two parts in a satisfying manner. The legs we have chosen for The Old Fisherman have a different sculpture style, but again, our merging process blends in these two parts nicely. In our results, the choice of the complementary statues may not correspond to the style of original statues, the complementary statues would naturally be better chosen by an expert art restorer.



Figure 5.4 – Registration and restoration of 5 incomplete statues, Esquiline Venus (first row), Old Fisherman (second row), Venus de Milo (third row), Wounded Amazon (fourth row) and Spinario (fifth). First column: initial point set, second column: overlay of the registered model and the point set, third column: final restoration.

CONCLUSION

This thesis developed a unified framework of virtual restoration of a statue that can keep the scanners' precision. We introduced a FAKIR algorithm that allows the registration of an anatomical model on a digital statue and identifies the statute point set's pose and anatomy. The obtained results are of high quality, both on synthetic data and on real data. We also presented two new skinning approaches to modify the position of a statue point set. Based on these contributions, we came up with a method to combine different statues after bringing them back to a common pose and morphology.

We have already tested proposed approaches to different datasets. It will be interesting to show the results of real statues' scans from the museum. We can organise the scanning event at Lugdunum museum in Lyon, one of our project members. The introduced virtual restoration framework gives restorers the possibility of testing different hypotheses before a real intervention. In addition, we can develop a user-friendly interface so that users can choose candidate statues, control the restoration parameters and compare different restoration hypotheses.

BIBLIOGRAPHY

- [AEa*11] ANTLEJ K., ERIC M., AVNIK M., UPANEK B., SLABE J., BATTESTIN B. B.: Combining 3D Technologies in the Field of Cultural Heritage: Three Case Studies. In *VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage - Short and Project Papers* (2011).
- [AHLG*13] ALI-HAMADI D., LIU T., GILLES B., KAVAN L., FAURE F., PALOMBI O., CANI M.-P.: Anatomy transfer. *ACM TOG* 32, 6 (2013), 188:1–188:8.
- [AL11] ARISTIDOU A., LASENBY J.: Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models* 73, 5 (2011), 243–260.
- [Ale02] ALEXA M.: Linear combination of transformations. *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002).
- [AMB*19] ALLDIECK T., MAGNOR M., BHATNAGAR B. L., THEOBALT C., PONS-MOLL G.: Learning to reconstruct people in clothing from a single RGB camera. In *CVPR* (2019).
- [AMX*18] ALLDIECK T., MAGNOR M., XU W., THEOBALT C., PONS-MOLL G.: Video based reconstruction of 3d people models. In *CVPR* (2018). CVPR Spotlight Paper.
- [ARM*19] ANGLES B., REBAIN D., MACKLIN M., WYVILL B., BARTHE L., LEWIS J., VON DER PAHLEN J., IZADI S., VALENTIN J., BOUAZIZ S., ET AL.: Viper: Volume invariant position-based elastic rods. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–26.
- [AS07] ANGELIDIS A., SINGH K.: Kinodynamic skinning using volume-preserving deformations. In *SCA '07* (2007).
- [ASC*13] ARBACE L., SONNINO E., CALLIERI M., DELLEPIANE M., FABBRI M., IDELSON A. I., SCOPIGNO R.: Innovative uses of 3d digital technologies to assist the restoration of a fragmented terracotta statue. *Journal of Cultural Heritage* 14 (2013), 332–345.

- [ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: shape completion and animation of people. In *ACM SIGGRAPH*. 2005, pp. 408–416.
- [Bas00] BASTIONI M.: Make human: Open source tool for making 3d characters. <http://www.makehumancommunity.org/>, 2000.
- [BBK08] BRONSTEIN A., BRONSTEIN M., KIMMEL R.: *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008.
- [BBLR15] BOGO F., BLACK M. J., LOPER M., ROMERO J.: Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *ICCV* (2015).
- [BK00] BARRON C., KAKADIARIS I. A.: Estimating anthropometry and pose from a single image. In *CVPR* (2000), vol. 1, pp. 669–676.
- [BKL*16] BOGO F., KANAZAWA A., LASSNER C., GEHLER P., ROMERO J., BLACK M. J.: Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV* (2016), pp. 561–578.
- [Blu67] BLUM H.: A Transformation for Extracting New Descriptors of Shape. In *Models for the Perception of Speech and Visual Form*, Wathen-Dunn W., (Ed.). MIT Press, 1967, pp. 362–380.
- [BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. *ACM TOG* 26, 3 (2007).
- [BS91] BLOOMENTHAL J., SHOEMAKE K.: Convolution surfaces. In *ACM SIGGRAPH* (1991), pp. 251–256.
- [CK10] CHAUDHURI S., KOLTUN V.: Data-driven suggestions for creativity support in 3d modeling. *ACM TOG* 29, 6 (2010), 183:1–183:10.
- [CKGK11] CHAUDHURI S., KALOGERAKIS E., GUIBAS L., KOLTUN V.: Probabilistic reasoning for assembly-based 3d modeling. *ACM TOG* 30, 4 (2011), 35:1–35:10.
- [CSWS17] CAO Z., SIMON T., WEI S.-E., SHEIKH Y.: Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR* (2017).
- [DL13] DIONNE O., LASA M.: Geodesic voxel binding for production character meshes. In *SCA '13* (2013).

- [ED17] ELNAGHY H., DORST L.: Geometry based faceting of 3d digitized archaeological fragments. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)* (2017), pp. 2934–2942.
- [FCD20a] FU T., CHAINE R., DIGNE J.: Anatomy changes and virtual restoration of statues. In *Eurographics Workshop on Graphics and Cultural Heritage* (2020).
- [FCD20b] FU T., CHAINE R., DIGNE J.: Fakir: An algorithm for revealing the anatomy and pose of statues from raw point sets. *Computer Graphics Forum* 39 (2020).
- [FDCP*08] FANTINI M., DE CRESCENZIO F., PERSIANI F., BENAZZI S., GRUPPIONI G.: 3d restitution, restoration and prototyping of a medieval damaged skull. *Rapid Prototyping Journal* 14 (09 2008), 318–324.
- [FKS*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. *ACM TOG* 23, 3 (2004), 652–663.
- [FTS08] FUNCK W. V., THEISEL H., SEIDEL H.: Volume-preserving mesh skinning. In *VMV* (2008).
- [GD96] GAVRILA D. M., DAVIS L. S.: 3-d model-based tracking of humans in action: a multi-view approach. In *CVPR* (1996), IEEE, pp. 73–80.
- [GPT03] GROSSMAN J. B., PODANY J., TRUE M.: History of the restoration of ancient stone sculptures.
- [GRP10] GILLES B., REVERET L., PAI D.: Creating and animating subject-specific anatomical models. *Computer Graphics Forum* 29, 8 (2010), 2340–2351.
- [GSDA*09] GALL J., STOLL C., DE AGUIAR E., THEOBALT C., ROSENHAHN B., SEIDEL H.-P.: Motion capture using joint skeleton tracking and surface estimation. In *CVPR* (2009), IEEE, pp. 1746–1753.
- [HBL*17] HUANG Y., BOGO F., LASSNER C., KANAZAWA A., GEHLER P. V., ROMERO J., AKHTER I., BLACK M. J.: Towards accurate marker-less human shape and pose estimation over time. In *3DV* (2017), IEEE, pp. 421–430.
- [HDD*92] HOPPE H., DEROSE T., DUCHAMP T., McDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *ACM SIGGRAPH* (1992), pp. 71–78.
- [HFG*06] HUANG Q.-X., FLÖRY S., GELFAND N., HOFER M., POTTMANN H.: Reassembling fractured objects by geometric matching. *ACM TOG* 25, 3 (July 2006), 569–578.

- [HLRB12] HIRSHBERG D. A., LOPER M., RACHLIN E., BLACK M. J.: Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *ECCV* (2012), Springer, pp. 242–255.
- [HSR*09] HASLER N., STOLL C., ROSENHAHN B., THORMÄHLEN T., SEIDEL H.-P.: Estimating body shape of dressed humans. *Computers & Graphics* 33, 3 (2009), 211–216.
- [HSS*09] HASLER N., STOLL C., SUNKEL M., ROSENHAHN B., SEIDEL H.-P.: A statistical model of human pose and body shape. In *Computer graphics forum* (2009), vol. 28, Wiley Online Library, pp. 337–346.
- [Hus55] HUS A.: Sculpture étrusque archaïque : le cavalier marin de la villa giulia. *Mélanges de l'école française de Rome* 67, 1 (1955), 69–126.
- [HWCO*13] HUANG H., WU S., COHEN-OR D., GONG M., ZHANG H., LI G., B.CHEN: L1-medial skeleton of point cloud. *ACM TOG* 32, 4 (2013), 65:1–65:8.
- [ICO64] ICOMOS: International charter for the conservation and restoration of monuments and sites. In *11th International Congress of Architects and Technicians of Historic Monuments, Venice* (1964), pp. 25–31.
- [IKKP17] ICHIM A. E., KADLECEK P., KAVAN L., PAULY M.: Phace: physics-based face modeling and animation. *ACM Trans. Graph.* 36 (2017), 153:1–153:14.
- [JBPSH14] JACOBSON A., BARAN I., POPOVIC J., SORKINE-HORNUNG O.: Bounded biharmonic weights for real-time deformation. *Communications of the ACM* 57 (2014), 99 – 106.
- [JCZ19] JIANG H., CAI J., ZHENG J.: Skeleton-aware 3d human shape reconstruction from point clouds. In *ICCV* (2019), pp. 5431–5441.
- [JLW10] JI Z., LIU L., WANG Y.: B-mesh: a modeling system for base meshes of 3d articulated shapes. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 2169–2177.
- [JMD*07] JOSHI P., MEYER M., DEROSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. *ACM TOG* 26 (2007), 71.
- [JS11] JACOBSON A., SORKINE O.: Stretchable and twistable bones for skeletal shape deformation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)* 30, 6 (2011), 165:1–165:8.

- [JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. *ACM TOG 24* (2005), 561–566.
- [KCKK12] KALOGERAKIS E., CHAUDHURI S., KOLLER D., KOLTUN V.: A probabilistic model for component-based shape synthesis. *ACM TOG 31*, 4 (2012), 55:1–55:11.
- [KCvO07] KAVAN L., COLLINS S., ŽÁRA J., O’SULLIVAN C.: Skinning with dual quaternions. In *I3D* (2007), ACM, pp. 39–46.
- [KCZO08] KAVAN L., COLLINS S., ZÁRA J., O’SULLIVAN C.: Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph. 27* (2008), 105:1–105:23.
- [KIL*16] KADLECEK P., ICHIM A.-E., LIU T., KRIVANEK J., KAVAN L.: Reconstructing personalized anatomical models for physics-based body animation. *ACM Trans. Graph. 35*, 6 (2016).
- [KJP02] KRY P., JAMES D. L., PAI D.: Eigenskin: real time large deformation character skinning in hardware. In *SCA '02* (2002).
- [KSH12] KAVAN L., SORKINE-HORNUNG O.: Elasticity-inspired deformers for character articulation. *ACM Trans. Graph. 31* (2012), 196:1–196:8.
- [Kv05] KAVAN L., ŽÁRA J.: Spherical blend skinning: A real-time deformation of articulated models. In *I3D* (2005), ACM, pp. 9–16.
- [LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *ACM SIGGRAPH* (2000), pp. 165–172.
- [LGK*12] LEE D., GLUECK M., KHAN A., FIUME E., JACKSON K.: Modeling and simulation of skeletal muscle for computer graphics: A survey. *Foundations and Trends® in Computer Graphics and Vision 7*, 4 (2012), 229–276.
- [LH16] LE B., HODGINS J.: Real-time skeletal skinning with optimized centers of rotation. *ACM Transactions on Graphics 35* (07 2016), 1–10.
- [LIPM19] LAZOVA V., INSAFUTDINOV E., PONS-MOLL G.: 360-degree textures of people in clothing from a single image. In *3DV* (2019), IEEE, pp. 643–653.
- [LL19] LE B. H., LEWIS J. P.: Direct delta mush skinning and variants. *ACM TOG 38*, 4 (2019), 113:1–113:13.

- [LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: Smpl: A skinned multi-person linear model. *ACM TOG* 34, 6 (2015), 1–16.
- [LRB*16] LITANY O., RODOLÀ E., BRONSTEIN A., BRONSTEIN M., CREMERS D.: Non-rigid puzzles. *Computer Graphics Forum* 35 (2016), 135–143.
- [LSS*19] LI C.-L., SIMON T., SARAGIH J., PÓCZOS B., SHEIKH Y.: Lbs autoencoder: Self-supervised fitting of articulated meshes to point clouds. In *CVPR* (2019), pp. 11967–11976.
- [MG03] MOHR A., GLEICHER M.: Building efficient, accurate character skins from examples. *ACM SIGGRAPH 2003 Papers* (2003).
- [MMRC20] MARIN R., MELZI S., RODOLÀ E., CASTELLANI U.: Farm: Functional automatic registration method for 3d human bodies. *Computer Graphics Forum* 39, 1 (2020), 160–173.
- [MTG03] MOHR A., TOKHEIM L., GLEICHER M.: Direct manipulation of interactive character skins. In *I3D* (2003), ACM, pp. 27–30.
- [MTLT88] MAGNENAT-THALMANN N., LAPERRIÈRE R., THALMANN D.: Joint-dependent local deformations for hand animation and object grasping. In *Proc. Graphics Interface '88* (1988), Canadian Information Processing Society, pp. 26–33.
- [MtLTM88] MAGNENAT-THALMANN N., LAPERRIRE R., THALMANN D., MONTRÉAL U. D.: Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface '88* (1988), pp. 26–33.
- [New34] NEWELL A. N.: Gallo-roman religious sculpture. *Greece and Rome* 3, 8 (1934), 74–84.
- [PF01] PLANKERS R., FUA P.: Articulated soft objects for video-based body modeling. In *ICCV* (2001), vol. 1, IEEE, pp. 394–401.
- [PIT*16] PISHCHULIN L., INSAFUTDINOV E., TANG S., ANDRES B., ANDRILUKA M., GEHLER P., SCHIELE B.: Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR* (2016), pp. 4929–4937.
- [PMG*05] PAULY M., MITRA N. J., GIESEN J., GROSS M., GUIBAS L. J.: Example-based 3d scan completion. In *SGP* (2005), Eurographics Association.
- [PMPHB17] PONS-MOLL G., PUJADES S., HU S., BLACK M. J.: Clothcap: Seamless 4d clothing capture and retargeting. *ACM TOG* 36, 4 (2017), 1–15.

- [PMRMB15] PONS-MOLL G., ROMERO J., MAHMOOD N., BLACK M. J.: Dyna: A model of dynamic human shape in motion. *ACM TOG* 34, 4 (2015), 120:1–120:14.
- [POC05] POLICARPO F., OLIVEIRA M. M., COMBA J. A. L. D.: Real-time relief mapping on arbitrary polygonal surfaces. In *I3D* (2005), ACM, p. 155–162.
- [RHC09] ROHMER D., HAHMANN S., CANI M.: Exact volume preserving skinning with shape control. In *SCA '09* (2009).
- [RTTP17] REMELLI E., TKACH A., TAGLIASACCHI A., PAULY M.: Low-dimensionality calibration through local anisotropic scaling for robust hand model personalization. In *ICCV* (2017).
- [SBB10] SIGAL L., BALAN A., BLACK M. J.: HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV* 87, 1 (2010), 4–27.
- [SBR*04] SIGAL L., BHATIA S., ROTH S., BLACK M. J., ISARD M.: Tracking loose-limbed people. In *CVPR* (2004), vol. 1, pp. 421–428.
- [SCYW15] SUN F., CHOI Y.-K., YU Y., WANG W.: Medial meshes—a compact and accurate representation of medial axis transform. *TVCG* 22, 3 (2015), 1278–1290.
- [SHG*11] STOLL C., HASLER N., GALL J., SEIDEL H.-P., THEOBALT C.: Fast articulated motion tracking using a sums of gaussians body model. pp. 951–958.
- [SRC01] SLOAN P., ROSE C. F., COHEN M.: Shape by example. In *I3D '01* (2001).
- [SZK15] SAITO S., ZHOU Z.-Y., KAVAN L.: Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Trans. Graph.* 34, 4 (2015).
- [TCSP04] THALMANN N., CORDIER F., SEO H., PAPAGIANNAKIS G.: Modeling of bodies and clothes for virtual environments. vol. 0, pp. 201–208.
- [TDS*16] TAGLIASACCHI A., DELAME T., SPAGNUOLO M., AMENTA N., TELEA A.: 3d skeletons: A state-of-the-art report. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 573–597.
- [TGB13] THIERY J.-M., GUY É., BOUBEKEUR T.: Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM TOG* 32, 6 (2013), 1–12.
- [TPT16] TKACH A., PAULY M., TAGLIASACCHI A.: Sphere-meshes for real-time hand modeling and tracking. *ACM TOG* 35, 6 (2016), 222:1–222:11.

- [TTR*17] TKACH A., TAGLIASACCHI A., REMELLI E., PAULY M., FITZGIBBON A.: Online generative model personalization for hand tracking. *ACM TOG 36* (2017), 1–11.
- [TZMS04] THEOBALT C., ZIEGLER G., MAGNOR M., SEIDEL H.-P.: Model-based free-viewpoint video: Acquisition, rendering, and encoding. *Picture Coding Symposium* (2004), 1–6.
- [VBG*13] VAILLANT R., BARTHE L., GUENNEBAUD G., CANI M.-P., ROHMER D., WYVILL B., GOURMEL O., PAULIN M.: Implicit skinning: real-time skin deformation with contact modeling. *ACM TOG 32*, 4 (2013), 125:1–125:12.
- [WC91] WANG L.-C., CHEN C.-C.: A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation 7*, 4 (1991), 489–499.
- [WHB11] WEISS A., HIRSHBERG D., BLACK M. J.: Home 3d body scans from noisy image and range data. In *ICCV* (2011), pp. 1951–1958.
- [WP02] WANG X. C., PHILLIPS C.: Multi-weight enveloping: Least-squares approximation techniques for skin animation. In *SCA* (2002), ACM, pp. 129–138.
- [WPP07] WANG R. Y., PULLI K., POPOVIC J.: Real-time enveloping with rotational regression. In *SIGGRAPH 2007* (2007).
- [WZC12] WEI X., ZHANG P., CHAI J.: Accurate realtime full-body motion capture using a single depth camera. *ACM TOG 31*, 6 (2012), 1–12.
- [YZZ*19] YU T., ZHENG Z., ZHONG Y., ZHAO J., DAI Q., PONS-MOLL G., LIU Y.: Simulcap: Single-view human performance capture with cloth simulation. In *CVPR* (2019), IEEE, pp. 5499–5509.
- [ZB15] ZUFFI S., BLACK M. J.: The stitched puppet: A graphical model of 3D human shape and pose. In *CVPR* (2015), pp. 3537–3546.
- [ZPBPM17] ZHANG C., PUJADES S., BLACK M. J., PONS-MOLL G.: Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *CVPR* (2017), pp. 4191–4200.

LIST OF PUBLICATIONS

- [FCD20a] FU T., CHAINE R., DIGNE J.: Anatomy Changes and Virtual Restoration of Statues. In *Eurographics Workshop on Graphics and Cultural Heritage* (2020).
- [FCD20b] FU T., CHAINE R., DIGNE J.: Fakir: An algorithm for revealing the anatomy and pose of statues from raw point sets. *Computer Graphics Forum 39* (2020).

LIST OF FIGURES

1	FAKIR is able to detect the morphology and pose of articulated shapes, given an elementary anatomical model, from a single static scan. It is equally effective for human, animal shapes and even imaginary creatures.	13
2	Restoration and pose change of the Prince Paris statue. Through articulated model regression, we identify the anatomy and pose. We then adapt parts from other statues after a change of morphology and use them to complete the input. Once the statue is complete, we can also change its pose: here we bring it to the pose of <i>The Thinker</i> by Auguste Rodin.	15
1.1	Restoration by 3d printing technology	17
1.2	Virtual restoration by reassembling fractured pieces	18
1.3	Diana, goddess of the hunt. It is restored in 2013.	19
1.4	Two human models in different shapes and poses	20
1.5	Examples of shape representation based on volumetric primitives	20
1.6	Different methods to discover the skeleton.	21
1.7	Different registration methods.	22
1.8	Comparaison of linear and dual quaternion blending. Images from [KCZO08]	23
2.1	Examples of statues with unrealistic anatomies.	26
2.2	The sphere-mesh of a bone is the union of the spheres centered on segment $[c_1c_2]$, with radius varying linearly between the two extremities of the segment.	27
2.3	Skeleton and sphere-mesh models for humans and centaurs. For the human model: the bones are organized into 5 chains shown in different colors. 4 additional bones are drawn in black: the pelvis which is a constrained triplet of bones, and the connection bone between the spine and the neck. Our model can also be adapted to various creatures, even imaginary ones such as a centaur.	28
2.4	Various projection cases. p has two possible projections \tilde{p}_α and $\tilde{p}_{-\alpha}$ depending on the orientation of the normal at p . Point p^* is the projection of p on line c_1c_2 . If the normal at p is oriented upward $\tilde{p} = \tilde{p}_\alpha$. Otherwise, $\tilde{p} = \tilde{p}_{-\alpha}$. The same strategy is used to project points q and o	29

2.5 Pairwise Optimization. With fixed extremities c_k and c_{k+2} , the pair of bones B_k and B_{k+1} is first rotated around axis $c_k c_{k+2}$ in order to minimize the two-bones energy. Then the lengths of the bones B_k and B_{k+1} and their common radius r_{k+1} are optimized successively. After these updates, the point-to-bone assignment is recomputed. As the process is repeated the distances are more accurate since the point-to-bone assignment becomes more meaningful. 33

2.6 Overview of the forward and backward iterative registration for a 3-bone chain. From an initial position (a), the chain extremity c_1 is fixed and the first bone B_1 is rotated and scaled to roughly calibrate its dimensions and pose through the optimization of the one-bone energy (b); the bone B_1 is fixed and the parameters of the second bone B_2 are roughly calibrated in turn (c); joint c_2 which is common to the first two bones is scaled and its position is optimized, by using the two-bones energy, the other joints being fixed (d); The position and length of the third bone B_3 are then coarsely calibrated through one-bone optimization and the process continues by alternating single bone optimization and two-bones optimization, until the last bone of the chain (e). After this first coarse calibration forward pass finishes, a backward pass using only two-bone optimizations is performed (f) permitting to refine the pose and skeleton parameters and solve for the chain extremity position. With few forward and backward pass involving two-bone optimization only, the model is registered (g). 36

2.7 The assigned points for the first spine is showed in red. The spine is first registered with points that are not yet assigned to the legs, which distorts its position. Once the legs are registered, the registration backtracks to the spine and its position can be corrected. The final result is shown on Figure 2.16. 38

2.8 Evolution of the registration distance with the iterations for different number of points in the point set (left image), different levels of Gaussian noise and Poisson noise (right image). 40

2.9 Evaluation of FAKIR with respect to increasing Gaussian noise after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain. From left to right: without noise, $\sigma = 0.5$, $\sigma = 1$, and $\sigma = 2$. The total groundtruth model length is 140 (All values are given in length units). . . . 40

2.10 Evaluation of FAKIR with respect to increasing Poisson noise after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain. From left to right: without noise, $\lambda = 0.5$, $\lambda = 1$, and $\lambda = 2$. The total groundtruth model length is 140 (All values are given in length units). . . . 41

2.11	Evaluation of FAKIR with respect to a bad initial anchor point position after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain. The last column shows that due to a bad initialization, the points (plotted in red) that are affected to the first bone do not bring enough information for the one-bone energy to move the chain extremity. Then, not enough bones remain to approximate the whole point set.	42
2.12	Evaluation of the FAKIR algorithm with respect to missing data after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain.	42
2.13	Registration of 6 statues: the Dancer with Crotales (first row at left), the Goddess Parvati(first row at right), the Saint Joint the Baptist (second row at left), the Aphrodite (seconde row at right), the Dancing Faun (third row at left) and the Age of Bronze (third row at right). For each result, we show from left to right: initial point set, overlay of the registered model and the point cloud, registered model.	46
2.14	Registration of 4 incomplete statues, Esquiline Venus (first row), Old Fisherman (second row), Venus de Milo (third row) and Wounded Amazon (fourth row). First column: initial point set, second column: overlay of the registered model and the point set, third column: registered model alone.	47
2.15	FAKIR result on the Mermaid statue, the skeleton is a human one but with a single chain instead of two legs.	48
2.16	Skeleton registration on human in various poses from the TOSCA dataset.	49
2.17	Skeleton registration on different animals or created characters from the TOSCA dataset. The skeleton is simply the human skeleton (Figure 2.3) supplemented with a bone chain for the tail.	50
2.18	ℓ^1 -medial skeletons [HWCO*13] extracted from the Aphrodite and Danseuse with Crotales point sets. To compare to our results on Figures 2.19 and 2.20.	51
2.19	Comparison with Pinocchio [BP07] and FARM [MMRC20] algorithm on the Aphrodite statue. From left to right: (a) Pinocchio with the Pinocchio-provided initial skeleton (17 bones); (b) Pinocchio with our initial skeleton (22 bones); (c) FAKIR with our initial skeleton after a single forward iteration; (d) FAKIR with our initial skeleton in 10 iterations. Only the skeleton is displayed since the bone radii are not taken into account by Pinocchio. (e) Model reconstructed by the FARM method.	51

2.20	Comparison with SMPLify on the Danseuse with Crotales. (a) FAKIR registration; (b) SMPLify using DeepCut predicted 3D joint positions on a single rendering; (c) SMPLify using DeepCut predicted 3D joint positions on two rendered views; (d) SMPLify using 3D joint positions estimated by FAKIR; (e) FARM registration. (First and third column: overlaid registered model; Second and fourth column: registered model)	52
2.21	Simultaneous Optimization for a chain of bones applied after the first Fakir forward pass. From left to right: initial position (after our forward step), position after 10 iterations, position after 35 iterations, position after convergence (43 iterations).	53
2.22	Simultaneous Optimization for a chain of bones applied directly from the initial position.	53
2.23	From left to right: Aphrodite statue, registration result without the normal-constrained projection, registration result with our normal-constrained projection	53
2.24	From left to right: the Dancer with Crotales statue, initialization position at bounding box center of point cloud, automatic registration result.	54
2.25	A failure case for automatic initialization. From left to right: the Victoria from TOSCA data set, initialization position at bounding box center of point cloud, automatic registration result.	54
3.1	Pose change for a mesh with folded surface sheets by heightfield skinning.	57
3.2	Heightfield skinning. From left to right: original point set; pose change with direct point set skinning; pose change with heightfield skinning.	58
3.3	Skinning weights for the bending rotation around a joint. (a) the red curve represents the influence weights of the left bone B_k and the blue curve represents the influence of the right bone B_{k+1} . The influence area of each bone is controlled by ε_r . (b): values of anisotropic ε_r on a bone. Points such that $\varepsilon_r = 0$ are shown in white. From yellow to red, the value of ε_r increases linearly.	59
3.4	Influence of ε_r on the skinning resulting from a bending rotation. In the first figure, $\varepsilon_r = 0$. The value ε_r increases from left to right.	60
3.5	Comparison of our anisotropic skinning method with Linear Blend Skinning and Dual Quaternions for a bending rotation. For a fair comparison, the two first methods use the Gaussian weight of Equation 3.1 which is made anisotropic in our anisotropic skinning. Dual Quaternions fix the volume collapse of Linar Blend Skinning near the convexity, but artefacts remain in the concave part, while our skinning method does not suffer from any of these flaws.	60

3.6	Twist motion: near blend Skinning and Dual-Quaternions Skinning to be compared with our approach in figure 3.7 that uses twist specific weights. Green dots show points that were aligned before the twisting rotation.	61
3.7	The blue curve represents the value of ω_β . Points v_{k+1} and w_{k+1} define the range of the twisting effect of B_{k+1} . p^* is the projection of p on the skeleton line. . . .	62
3.8	Comparison of skinning approaches on a sphere-mesh model with combined twist and bend motions. Left to right: Isotropic Linear Blend Skinning, Isotropic Skinning with Dual Quaternions, Ours	62
3.9	Registration and pose changes of 4 statues: the Dancer with Crotales (first row), the Dancing Faun (second row), Aphrodite (third row) and the Old Man Walking (fourth row). First column: initial point set, second column: overlay of the registered model and the point cloud, third column: final point set in a modified pose by our skinning method, fourth column: skinning result with Dual Quaternion. . .	63
4.1	Illustration of baseline and base-point	67
4.2	Illustration of baseline before and after a movement	68
4.3	Illustration of baseline and base-point after transformation	68
4.4	Intersection of two cones	72
4.5	Cutting planes and intersection curves on our model	73
4.6	The positions of segments on a baseline before and after a movement.	74
4.7	The positions of segments on a baseline before and after a movement.	74
4.8	Baseline skinning on synthetic data for a folding movement (first row) and an unfolding movement (second row). From left to right: initial position, linear interpolation for α_p and α_d , cubic interpolation for α_p and α_d	75
4.9	Skinning results	77
5.1	Restoration and pose change of the Prince Paris statue. Through articulated model regression, we identify the anatomy and pose. We then adapt parts from other statues after a change of morphology and use them to complete the input. Once the statue is complete, we can also change its pose: here we bring it to the pose of <i>The Thinker</i> by Auguste Rodin.	79
5.2	The importance of morphology change in restoration. Left: restoration with scaling morphology of the leg part. Right: restoration using directly the original leg part.	81
5.3	Overlap area. Left: blue points and red points come from different statues. Right: Merging result.	81

- 5.4 Registration and restoration of 5 incomplete statues, Esquiline Venus (first row), Old Fisherman (second row), Venus de Milo (third row), Wounded Amazon (fourth row) and Spinario (fifth). First column: initial point set, second column: overlay of the registered model and the point set, third column: final restoration. 83

