

FAKIR : An algorithm for estimating the pose and elementary anatomy of archaeological statues

TONG FU, LIRIS, CNRS, Université de Lyon
 RAPHAËLLE CHAINE, LIRIS, CNRS, Université de Lyon
 JULIE DIGNE, LIRIS, CNRS, Université de Lyon

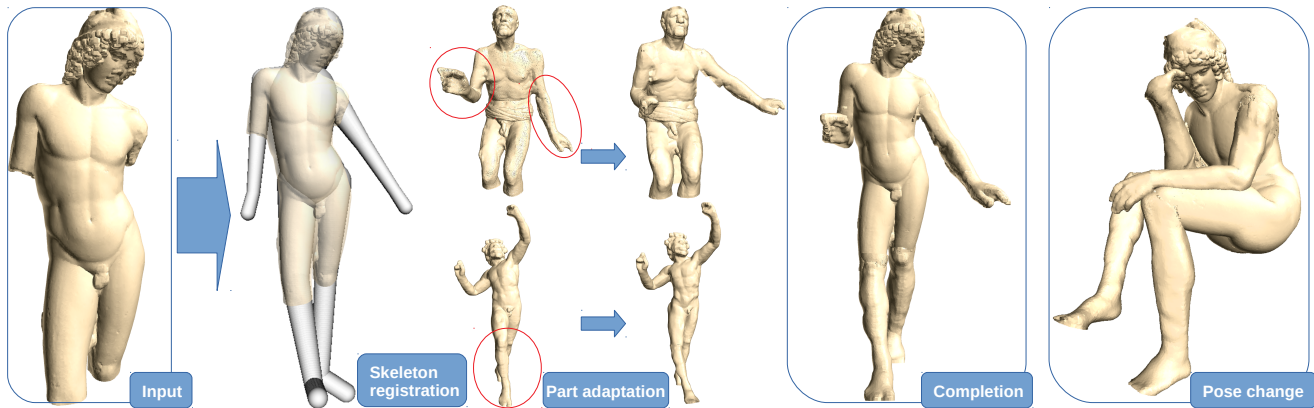


Fig. 1. Application of our method to pose change and shape completion on the Prince Paris statue. Through skeleton regression on statues, we identify their pose and anatomy. If the input statue has missing data, we propose to adapt parts from other statues and use them to complete the input. Once the statue is complete, we can also change its pose: here we bring it to the pose of *The Thinker* by Auguste Rodin.

The digitization of archaeological artefacts has become an essential part of cultural heritage research be it for purposes of preservation or restoration. Statues, in particular, have been at the center of many projects. In this paper, we introduce a way to improve the understanding of acquired statues by registering a simple and pliable anatomical model to the raw point set data. Our method performs a Forward And backWard Iterative Registration (FAKIR) which proceeds joint by joint, needing only a few iterations to converge. Furthermore, we introduce a simple detail-preserving skinning approach working directly on the point cloud, without needing a mesh. By combining FAKIR with our skinning method we are able to detect the pose and the elementary anatomy of a sculpture and modify it, paving the way for pose-independent style comparison and statue restoration by combination of parts belonging to statues with different poses.

CCS Concepts: • **Computing methodologies** → **Shape modeling; Point-based models.**

Additional Key Words and Phrases: Skeleton registration, point set analysis, anatomy detection, shape editing

1 INTRODUCTION

With the progress of 3d scanning techniques, it is now common to create digital replicas of artworks, which will remain forever intact, while the real-world counterparts will slowly decay due to time damage or human activity. However when the digitization is performed, the statues are often already degraded. Restorations being costly, invasive and sometimes even risky, museums are often

reluctant to carry out such processes. Hopefully, digitization offers another huge advantage: the possibility to build and test restoration hypotheses, whether or not they are applied to the real model afterwards. In the statue's case, one of the first steps towards virtual restoration is to be able to identify anatomical parts, in order to guide the restoration. While this task can be performed manually, it is often long and tedious. In this paper, we are thus interested in recovering the anatomy of a statue, using a simple anatomical model, while keeping user interventions to a minimum. We focus on human statues with no or few garments, since those will benefit directly from a registered anatomical model. Many Roman or Gallo-Roman statues fall within this scope. Furthermore, we consider that the digitized statues are provided as point sets.

Our objective is to identify the elementary anatomy of a statue allowing in turn to change the statue's pose. To do so, we first propose a method for calibrating and registering a simple anatomical model to a point set. This step is achieved directly on the point cloud, avoiding thus the tedious meshing step and preserving the accuracy of the initial sampling. To perform the calibration and registration, we introduce the Forward And backWard Iterative Registration (FAKIR) algorithm, inspired by recent inverse kinematics approaches. FAKIR permits to efficiently register the anatomical model in only a few iterations. Once the model is registered, the point set surface is locally represented as a residual heightfield above the registered anatomical model. It is then possible to change the pose of the statue and combine parts of different statues after they are brought to a

Authors' addresses: Tong Fu, tong.fu@liris.cnrs.fr LIRIS, CNRS, Université de Lyon; Raphaëlle Chaine, raphaelle.chaine@liris.cnrs.fr LIRIS, CNRS, Université de Lyon; Julie Digne, julie.digne@liris.cnrs.fr LIRIS, CNRS, Université de Lyon.

common pose. To do this, we propose a skinning approach that is applied directly to the point cloud, without using a mesh.

To summarize, our contributions are the following:

- A simple anatomical model efficiently representing a statue pose.
- An efficient calibration and registration process based on inverse kinematics.
- A point set skinning process permitting to modify the point set following an anatomy and/or pose change.

2 RELATED WORK

Anatomical Model. Designing anatomical models for human shapes has raised a lot of interest. The most common representation consists in a more or less detailed graph of bones such as the ones used in the MakeHuman framework [5]. While some methods go beyond the human skeleton representation and model every single muscle to increase realism [21], we will focus here on skeletons, which are simple and efficient enough for our purpose. Among skeleton-based models, the sphere-mesh model [30], a variant of convolution surfaces [6], has been introduced for representing mesh models by packing sphere into it and encoding its structure. Conceptually, the sphere-mesh model can be seen as a piecewise linear simplification of the computational geometry skeleton [29]. Although the original sphere-mesh construction proposed to discover entirely the shape structure from an input mesh, the sphere-mesh model can be used to represent an anatomical model by imposing constraints on it. As such, it has been used successfully for representing hand skeletons [28, 31]. This model is light and pliable and we will also rely on it.

Skeleton rigging and skinning. Once a skeleton model is chosen, the next problem for animation purposes is to position it inside an input mesh and to attach surface points to it, processes that are called *rigging* and *skinning* respectively. Skeleton rigging can be performed manually, but a few methods have investigated automatic processes. In particular, the Pinocchio algorithm [4] adapts a skeleton to a static mesh by defining an objective function and maximizing it. It works by packing spheres into the mesh and by considering their centers, gathered in a graph, as the admissible joint positions. This pre-computation makes the skeleton pose estimation tractable. If the input data is dynamic, it is possible to infer, or track, a skeleton from it. Most tracking approaches focus on the direct independent and simultaneous capture of the positions of the joints, using a temporal sequence and prior constraints. The pose parameters (angles) and intrinsic parameters (*e.g.* bone lengths) are then inferred from it. Many of such tracking methods work with depth streams or videos starting from a previously calibrated skeleton but the calibration itself can be performed from a depth video and a set of known admissible poses [28, 31]. Such methods require a dynamic scene and cannot apply to the static mesh rigging problem. It is also possible to rely on a whole database of people scans to learn the pose and deformation of human bodies [2, 14, 15, 23]. Finally, some methods [13, 35] aim at finding a person’s pose despite its sometimes loose clothing, but this is outside the scope of our paper. For the sake of completeness it is also important to mention that many methods perform human tracking without requiring a model using multiple view acquisitions with or without markers. These methods are only

remotely linked to our problem and we refer the reader to [7] for an excellent overview.

Once the skeleton is correctly positioned, skinning methods aim at attaching the surface model to it by using weights that define the influence of the bones on the position of surface points. Then, when changing the pose of the skeleton, the attached surface should deform accordingly. Setting the right weights is an important question: while the profile of the weights is generally sketched by graphic designers [25], there exist automatic weighting techniques that, for example, use heat diffusion [4]. As far as skinning techniques are concerned, Linear Blend Skinning [24] is one of the most popular ones. A mesh surface point is transformed by a linearly weighted combination of the motion of the bones it is attached to. Several methods attempt to fix the well-known collapsing problem of Linear Blend Skinning, such as Pose Space Deformation [22], Multi-Weight Enveloping [34], spherical Skinning [19] and Dual-Quaternions Skinning [18]. A recent skinning method [20] 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 our anatomical model. Taking a different perspective on the problem, Implicit skinning [32] uses an implicit formulation of the surface that better supports pose changes and reprojects skinned vertices on the implicit model after each pose change. In this paper, we also use a proxy model but it is explicit.

Pose change. When a model is rigged and skinned, it is possible to change its pose manually by interacting with some joints of the skeleton. 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 frame of an animation. As a consequence, research has focused on inferring the motion from some key joints and frames with given skeleton positions. In this inverse kinematics context, the Fabrik [3] and CCD [33] 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. Our registration method will also use kinematic chains in a forward and backward approach, but the similarity ends here, since our goal is to estimate not only the pose but also length and width of the model limbs using data-attachment constraints in a static framework. Quite differently, some approaches transfer the volume delimited by a mesh to the interior of another mesh by minimizing some harmonic energy [1]. To do this, a deformation field is set up between the two meshes, using the correspondence with the nearest point in the current iteration [12].

Shape Synthesis. Shape synthesis is a very active research area. It is often done by reusing parts of existing models. [11] proposed an example-based modeling system, where new models are generated by stitching together statue parts from a database constructed by segmenting a set of input models. For completing a 3D model, a general idea is to retrieve suitable models from the database and warp the retrieved models to conform with the incomplete model [9, 27]. More recently, a probabilistic representation for the components

of a shape has been developed to suggest relevant components during an interactive assembly-based modeling session [8, 17]. We will propose an application to statue synthesis by part combination, but the choice of the parts is done manually while the part adaptation is automatic.

3 ANATOMY AND POSE ESTIMATION

3.1 Human model

Our work focuses on artworks representing human beings without or with only a few garments. A challenge of artistic data compared to human scans lies in the difference in aesthetic perception. Indeed many sculptors favored the perceived beauty of their work over the realism of human proportions [16, 26]. Figure 2 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 modeling every single limb in a very realistic way are too constrained for our purpose. In particular, in our model we avoid modeling muscles. After we have registered our simplified human model to the statue point cloud, a further step could be to add new flexible muscle models, but this goes beyond the scope of this paper 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 (Figure 2b).



(a) A Gallo-Roman statue of a Gallic warrior (Avignon, France, picture: F. (1530-1534) by Baccio Bandinelli, Philibert-Caillat) (b) Statue of Heracles and Cacus warrior (Florence, Italy, picture: Cyberuly)

Fig. 2. Examples of statues with unrealistic anatomies.

We introduce an anatomical model inspired by the sphere-mesh model [30], already successfully used for hand tracking [28, 31], 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 3b). 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 3a. Importantly 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*.

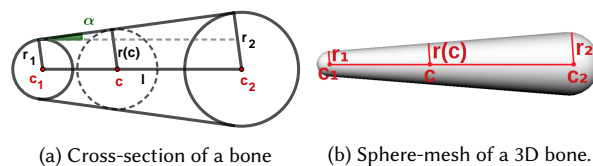


Fig. 3. The sphere-mesh of a bone is the union of the spheres centered on segment $[c_1c_2]$, with radius varying linearly between the two extremities.

With this type of bone element, we construct a simple human body template with a very coarse respect of human proportions as an initial body shape (Figure 4). Our human body template contains 22 bones $\{B_k\}_{k=1..22}$. Three of those 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, a special bone is used 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 5 chains, depicted in different colors in figure 4a: 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 chain organization is used to define the notion of predecessor and successor for one bone in a chain, and this ordering will be extensively used in our kinematic registration. In particular, we will reverse the ordering of the chain to process it forward and backward several times along the process. Each bone is thus fully defined by its intrinsic parameters (length and two radii) and by 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.

3.2 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 the limbs they should be attached to. We assume that the coordinates of the points are provided with a coarse approximation of the normal so that we can locally distinguish the inside from the outside of the sampled surface. If normals

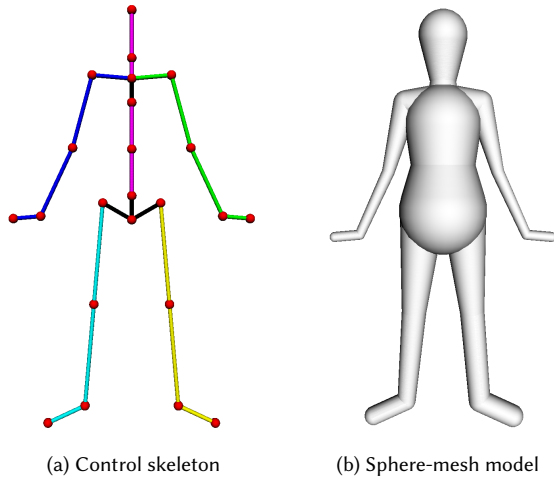


Fig. 4. Our anatomical human model: the bones are organized into 5 chains shown in different colors. 4 additional bones are drawn in black: the pelvis which is a constrained triplet of bones, and the connection bone between the spine and the neck.

are unavailable, in most cases, they can be roughly estimated and orientated using viewing directions from the 3D sensor for example.

Distance from a point to a one-bone model. We start by defining the normal-constrained projection of a sampled point p on a single bone B by using the normal vector \mathbf{n}_p to disambiguate the choice between several orthogonal projection possibilities. The sphere-mesh model calibration and registration strives to reduce the distance between the sampled points and their corresponding points on the sphere-mesh and using the normal orientation helps this process. 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. We abusively call these intersection points orthogonal projections (see appendix A for the detailed cases). From all these possibilities, 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 to better handle the case where the bone is not initialized close to the point set. Since each point p has a normal-constrained projection on several 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 associated closest bone.

Distance from a point set to a sphere-mesh model. Given a point set P and a sphere-mesh model of K bones, we first need to approximate the subset of corresponding points for 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 $d_k = \|p - \tilde{p}_k\|$, $k = 1 \dots K$. However, we try to favor the assignation to the conical part of a bone. Therefore, if p projects on the spherical part of the closest bone and if it also projects on the conical part of another bone with a similar distance, we assign it to this second bone. 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 :

$$E_k(P_k, B_k(l_k, \mathbf{r}_k), \boldsymbol{\theta}_k) = \sum_{p \in P_k} \|p - \tilde{p}_k\|^2 \quad (1)$$

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 previous bones in the chain.

In the following, the sum of one-bone distance functions is considered as an energy that we aim to minimize in order to capture the anatomy and the pose of the sphere-mesh that best correspond to our point set:

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

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 distance:

$$E_{k,k+1} = \sum_{p \in P_k} \|p - \tilde{p}_k\|^2 + \sum_{p \in P_{k+1}} \|p - \tilde{p}_{k+1}\|^2. \quad (3)$$

3.3 FAKIR : Forward And backWard Iterative Registration

To register our anatomical model to the acquired static point cloud, we propose a kinematic approach taking into account the articulated property of the skeleton. Contrarily to many tracking methods that exploit the redundancy of information between several frames or several views [28, 31], our method requires only one joint of the skeleton to be close to its optimal position, the rest of the skeleton pose being arbitrary, except for a mild assumption that in practice means using oversized initial lengths for the bones. It is one more assumption than the Pinocchio algorithm [4], which also works in the static case and starts with no prerequisite and we will compare our algorithm to it. Inspired by the FABRIK [3] and CCD [33] algorithms, our registration algorithm successively loops forward and backward through the chains of bones so as to rotate and rescale 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). An originality of our method is that bones are not only considered one by one but also by consecutive pairs, which allows for a more robust estimation of the pose and skeleton parameters along a chain.

Registration process for a chain of bones. If the bone B_k is close to the points to which it should be associated, the estimation of the one-bone energy function becomes meaningful and the minimization of this energy can be used to coarsely optimize the position and radii of that bone with respect to the data. Therefore, our approach amounts to iteratively updating the rotation and intrinsic parameters of B_k until we catch a coarse estimation of P_k and B_k , one extremity of B_k being kept fixed. Our algorithm gradually rotates the current bone B_k with respect to its predecessor, updating P_k after each rotation, so that P_k gradually contains more relevant points. 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 approximate the

full length of a bone accurately and we use this energy only to obtain a first approximation of the registration for one bone. In an intertwined manner, a finer local registration procedure of our algorithm is performed each time two consecutive bones B_k and B_{k+1} have been processed. Its goal is to optimize the common joint position and radius while fixing the two other joint extremities. This optimization is performed by minimizing a two-bones energy function, defined above. 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 in order to further refine the joints positions and radii between pairs of consecutive bones, only using two-bones energies. Finally, the position of the last extremity of a chain is also optimized between each forward and backward step of the loop by optimizing the one-bone energy function. Each parameter optimization is thus made by minimizing an energy related to either one or two bones. The full process is summarized in Algorithm 1 and illustrated with a chain of three bones in Figure 5.

Note that during the process, two consecutive bones should not be enclosed in a same limb of the model point set, otherwise a joint might remain enclosed inside this limb, trapping the optimization in a local minimum. This can be avoided by introducing a condition on the initial bone lengths ensuring that the sum of the initial lengths of two consecutive bones is below the optimal length of each of the limbs they are close to. A practical trick is to slightly oversize the bones before starting the registration process.

Optimization of a single bone. As stated above, the optimization of the one-bone energy is only used for estimating the parameters of the extremities of a chain, or for the very first forward pass in Algorithm 1). After this step, the optimization of the position of a joint using a bone pair should be preferred because it is more precise.

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 function $E_k(P_k, B_k(l_k, r_k), \theta_k)$, where θ_k are the angles of rotation with respect to the predecessor 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_1c_2 , indeed the rotation around c_1c_2 is not considered since it leaves the bone unchanged. In the case of the minimization with respect to the vector of rotation angles θ_k , we iteratively try to replace the current angles θ_k with an update $\theta_k + \delta\theta_k$. At a minimum, $\nabla_{\delta\theta_k} E_k(P_k, B_k(l_r, r_k), \theta_k + \delta\theta_k) = \mathbf{0}$, and the value for $\delta\theta_k$ follows. The details for the damped least-squares estimation are provided in Appendix B.

Optimization for the 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 function. The two end-sphere centers being fixed (c_k and c_{k+2} in Figure 6), we first compute the optimal rotation of the two bones around axis c_kc_{k+2} , all other parameters being fixed. 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

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 function;
 - 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 function.
 - 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 function.
 - 18: **end for**
 - 19: Compute the length of the last bone and the radius of the last sphere with the one-bone energy function.
 - 20: **until** convergence
-

computed as $\hat{r}_{k+1} = r_{k+1} + \delta r$. The parameters optimization alternates with a recomputation of point sets P_k and P_{k+1} , which refines the point-to-bone assignment. Similarly to the single-bone case, the optimization is also performed using the Levenberg-Marquardt algorithm whose details are provided in Appendix C.

Full Skeleton Registration. Our anatomical model is composed of 5 chains, one of which is of particular importance: the spine chain which connects all other chains (the arms, and the legs through the pelvis block). We assume that the pelvis part of our model is initialized near the corresponding part of the point set, which requires a very limited user interaction - basically only one 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. The position of the pelvis is then revised during the registration of the spine chain. The registration order is the following: first the spine chain is registered, followed by each of the two leg chains and each of the two arm chains. When registering the arms and legs chains, the position for the joint attached to the spine or the pelvis remains fixed.

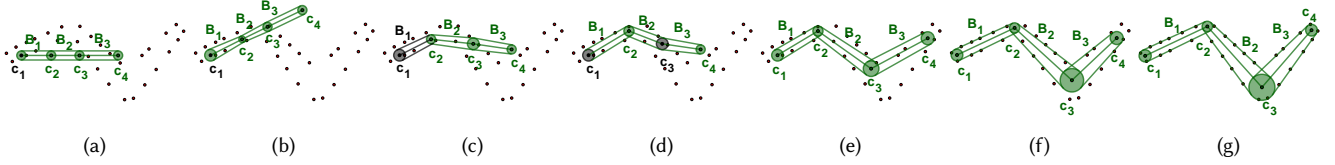


Fig. 5. Overview of the forward and backward iterative registration for a 3-bone chain. From an initial position (a), the chain extremity c_1 is fixed and the first bone B_1 is rotated and scaled to roughly calibrate its dimensions and pose through the optimization of a one-bone energy function (b); 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 a two-bones energy function, 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).

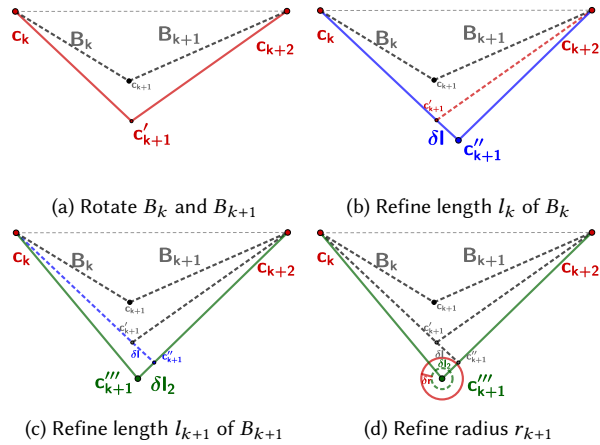


Fig. 6. Pairwise Optimization. With fixed extremities c_k and c_{k+2} , the pair of bones B_k and B_{k+1} is first rotated around axis $c_k c_{k+2}$ in order to minimize the two-bones energy. Then the lengths of the bones B_k and B_{k+1} and their common radius r_{k+1} are optimized successively. After 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.

4 POINT SET SKINNING AND APPLICATION TO STATUE POSE AND ANATOMY MODIFICATION

Once the anatomical model is registered to the statue point set, it is possible to exploit the assignment between the model and the point set to change the pose and the elementary anatomy of a statue by modifying the extrinsic and intrinsic parameters of the bones respectively. To do so we attach the point set to its registered anatomical model so that deforming the model will deform the point set adequately. This process, well known in the animation research community, is called *Skinning*. It allows to deform the skin following an underlying skeletal animation, a widespread method to animate 3D models. In most skinning methods, the skin is a 3D mesh and the skeleton is a tree whose nodes represent joints of the skeleton and edges represent bones. An originality of our approach 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. We rather propose to develop a point set skinning process.

In our case, since after applying FAKIR, the anatomical model is registered to the point set, it can serve as the skeleton of usual animation methods. We introduce a skinning method working on point sets and taking a better account of the twisting and bending rotations, alleviating known artefacts of Linear Blend Skinning and Dual Quaternions. We further propose to consider the skin of the model as a set of details, encoded as a heightfield on top of our elementary anatomical and transferred back on the model after pose or intrinsic parameters changes.

Local heightfield over the sphere-mesh model. From now on, 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} . Importantly enough, \tilde{p} is the regular orthogonal projection on the bone p is assigned to, in opposition to the normal-constrained projection defined in section 3.2 and appendix A. Hence the surface point set is decomposed into the set of base points on the sphere-mesh and the residual orthogonal heightfield. The pose and anatomy changes modify, through skinning, the position of the base point on the sphere-mesh model. Then the heightfield is added back to the modified projection yielding the final point set, as described in algorithm 2.

Under pose change, a point p remains affected to the same bone but its evolution depends on both the evolution of its base point \tilde{p} , the normal at \tilde{p} and the possible scaling of its the heightfield value $h(p)$. The evolution of the base point \tilde{p} is driven by the bone p is assigned to and its adjacent bones in the chain. The motion induced by our skinning model is continuous. Indeed, the position of a base point \tilde{p} continuously depends on a set of one to three consecutive bones. Although the point-to-bone assignment is fixed during deformation, the base point can slide along the 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 point is either preserved or continuously scaled, the whole motion is continuous throughout the deformation. Naturally, some

points may become hidden because of self-intersections between bones caused by the rotation. In this work, we choose to keep the self-intersections, since the resulting outer envelop of the model is visually satisfying. Compared to other heightfield decomposition methods, the heightfield is carried by the points themselves and not by the base surface. Hence it can encode folded surface sheets over the sphere-mesh model (see the *Dancer with Crotales* example on Figure 21). In the following we describe our proposed point set skinning method to be applied to the base point sets before lifting them with the heightfields (Algorithm 2). Since the skinning can be applied independently of the heightfield decomposition, we describe the method in a generic point set context for notation simplicity. Note however that heightfield skinning should be preferred to direct point set skinning, since it provides a much better result as illustrated on Figure 7.

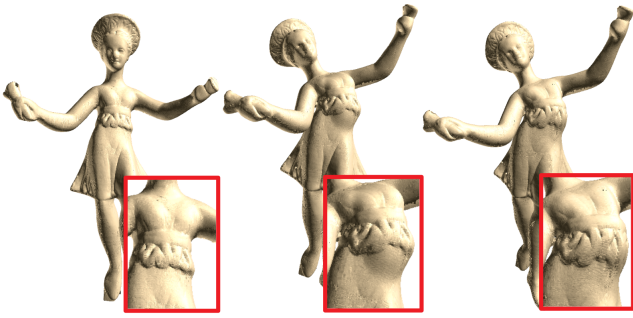


Fig. 7. Heightfield skinning. From left to right: original point set; pose change with direct point set skinning; pose change with heightfield skinning.

Algorithm 2 Height-field 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' .
-

4.1 Point Set Skinning

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 the point after a deformation is a weighted linear combination of the positions relative to its influencing bones. Linear Blend Skinning [24], 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 the joint is not preserved when it is bent around the axis orthogonal to the two adjacent bones (Figure 8). 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 12a). These flaws are avoided by Dual Quaternions Skinning [18] which interprets a combination of rigid transformations as a rigid transformation, however artefacts still appear in the concavities (Figures 11 and 14).

We propose to deal with the pose change in a different way that breaks down the movement into its natural components. 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. We also 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 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.

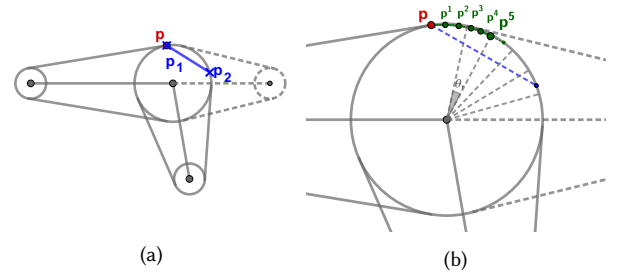


Fig. 8. Linear Blending artefacts: joint collapse (a); (b) trace of the evolution of a point through iterated re-skinning with direct linear blend skinning after splitting the rotation into smaller ones. Through Linear Blend Skinning, a point p located opposite to the bending is moved inside the joint along a line p_1p_2 where p_1 and p_2 are the initial positions of p with respect to bones B_1 and B_2 respectively.

Bending rotation with anisotropic weights. We decompose the bending rotation into a sequence of small bending rotations and update both the position through Linear Blend Skinning and the corresponding weights of the points after each rotation (Figure 8b). At each step, the bone is slightly rotated by θ_r around the joint bending axis and the points are moved following adapted Linear Blend Skinning weights. 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. In our experiments, we set ϵ_r larger than the average distance between the point set and our model. The weight $\omega_j(p)$, relative to one of p 's influencing bones B_j writes:

$$\omega_j(p) = \frac{1}{c} \exp - \frac{\|p\tilde{p}_j\|^2}{2(\epsilon_r/3)^2} \quad (4)$$

where c is a normalizing factor ensuring that the weights sum to 1, and \tilde{p}_j is the regular 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 9 shows the weights profile along two bones.

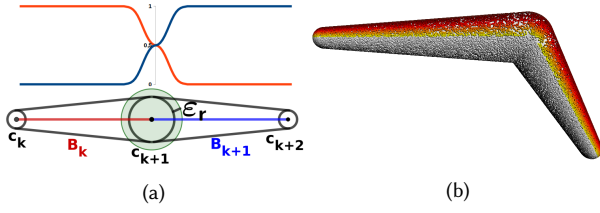


Fig. 9. Skinning weights for the bending rotation around a joint. (a) the red curve represents the influence weights of the left bone B_k and the blue curve represents the influence of the right bone B_{k+1} . The influence area of each bone is controlled by ϵ_r . (b): values of anisotropic ϵ_r on a bone. Points such that $\epsilon_r = 0$ are shown in white. From yellow to red, the value of ϵ_r increases linearly.

This discretization yields similar results as Dual Quaternion skinning. However, it is possible to further improve the method, by noticing that if ϵ_r is large, the convex part behaves well, it is populated by 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 10). Quite counter-intuitively, this self-intersection is much more visually satisfying than the thinning artefact which can be 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 [32]. 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 is larger for points on the convex parts. More precisely, ϵ_r varies continuously with respect to an *anisotropy angle*.

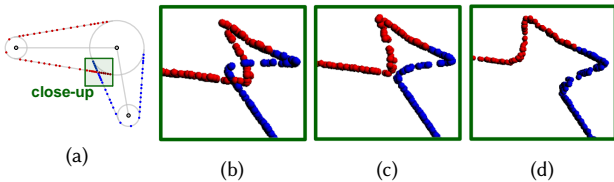


Fig. 10. Influence of ϵ_r on the skinning resulting from a bending rotation. In figure 10a, $\epsilon_r = 0$. The value ϵ_r increases from (b) to (d).

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 serves to transition continuously 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 $\epsilon_r = \cos \alpha_b \epsilon_r'$, 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:

$$\omega_j(p) = \begin{cases} \frac{1}{c} \exp -\frac{\|p\tilde{p}_j\|^2}{2(\cos \alpha_b \frac{\epsilon_r'}{3})^2} & \text{if } \cos \alpha_b < 0 \\ 1 & \text{if } \cos \alpha_b \geq 0 \text{ and } p \in P_j \\ 0 & \text{if } \cos \alpha_b \geq 0 \text{ and } p \notin P_j \end{cases} \quad (5)$$

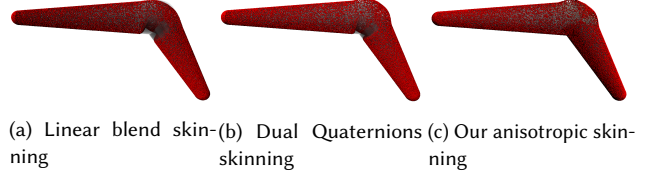


Fig. 11. Comparison of our anisotropic skinning method with Linear Blend Skinning and Dual Quaternions for a bending rotation. For a fair comparison, the two first methods use the Gaussian weight of Equation 4 which is made anisotropic in our case. Dual Quaternions fix the volume collapse of Linear Blend Skinning near the convexity, but artefacts remain in the concave part, while our skinning method does not suffer from any of these flaws.

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 drop the skinning framework by linear combinations of bone motions and replace it 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

$$\omega(p) = \frac{\|v_{k+1} - p^*\|}{\|v_{k+1} - c_{k+1}\| + \|c_{k+1} - w_{k+1}\|} \quad (6)$$

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 13). 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 12 and 13, we compare our method with Linear Blend Skinning and Dual Quaternions for a twisting rotation, the trace of the points initially aligned is much smoother with our approach.

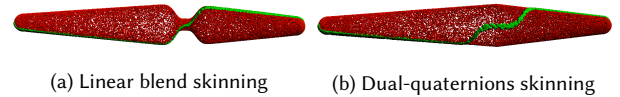


Fig. 12. Linear blend skinning and dual-quaternions skinning to be compared with our approach in figure 13 that uses twist specific weights. Green dots show points that were aligned before the twisting rotation.

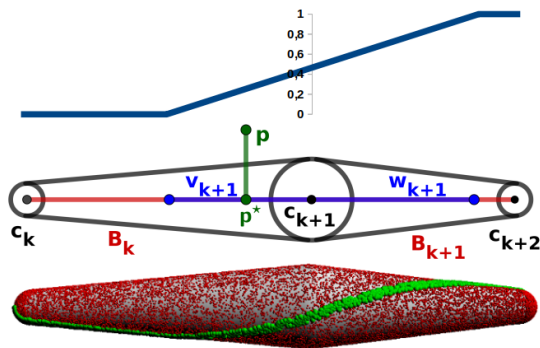


Fig. 13. The blue curve represents the value of ω_β . Points v_{k+1} and w_{k+1} define the range of the twisting effect. p^* is the projection of p on the skeleton line.

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 14 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 and our bending approach without anisotropy get a similar result, with a region of influence restricted to the neighborhood of the joint. 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.



Fig. 14. Comparison of skinning approaches on a sphere-mesh model with combined twist and bend rotation movements. Left to right: Linear Blend Skinning, skinning with Dual Quaternions, Ours

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 distance update parallelization. All experiments are run on an Intel Core i7-4790K CPU @ 4.00GHz.

5.1 Experiments on synthetic data

We first test our algorithm on synthetic data to provide a quantitative evaluation of the FAKIR performances. We consider a point set of $5k$ points sampled on a sphere-mesh of a 4-bone chain in a specific pose. We start from a generic 4-bone chain that we try to register to the point set. Although the point set and the initial chain are quite distant from each other, a user-given initial approximate 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:

$$dist = \frac{1}{N_{points}} \sum_{p \in P} \|p - \tilde{p}\|. \quad (7)$$

For $5k$ points, without any additional noise, our algorithm takes 5.2s to converge to $dist = 0$ in 7 iterations for this synthetic model of $K = 4$ bones, including 3.2s for the initialization. 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 15: 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. In fact, FAKIR is rather resilient to even relatively high levels of Gaussian noise (Figure 16). Figure 17 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 is rather robust, but in some cases (last column), when the initial chain position is such that the points corresponding to the first bone only project in a small area around the joint between the first and second bones. In this case, the optimization of the one-bone energy function 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 slightly modifying the one-bone energy function by adding a bone occupancy term. A preferential alternative would be to modify the one-bone energy function of the first and last bones by adding a term corresponding to the distance of the free caps to the data points. However, with our working assumptions those cases are avoided. FAKIR is also rather robust to missing data thanks to the iterated forward and backward passes (Figure 18). 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.

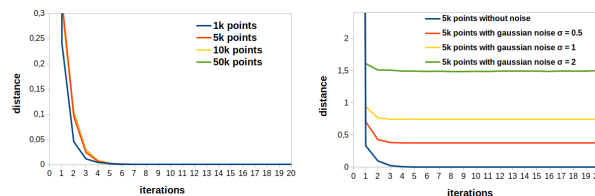


Fig. 15. Evolution of the registration distance with the iterations for different number of points in the point set (left image) and different levels of noise (right image).

5.2 Skeleton registration results on statues

We selected some interesting statues from various sources.

- (1) Dancer with Crotales: Louvre Museum, Paris, France.
- (2) Dancing Faun: Pompei excavations, Italy.
- (3) Aphrodite, Museum of Thorvaldsens, Copenhagen, Denmark.
- (4) Old man walking: Nye Carlsberg Glyptotek, Copenhagen, Denmark

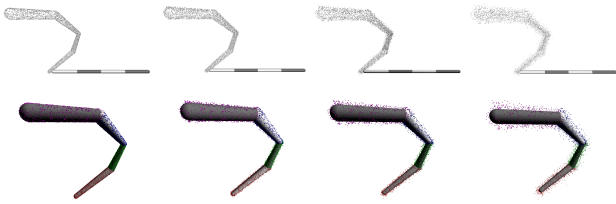


Fig. 16. Evaluation of FAKIR with respect to increasing Gaussian noise after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain. From left to right: without noise, $\sigma = 0.5$, $\sigma = 1$, and $\sigma = 2$. The total groundtruth model length is 140.

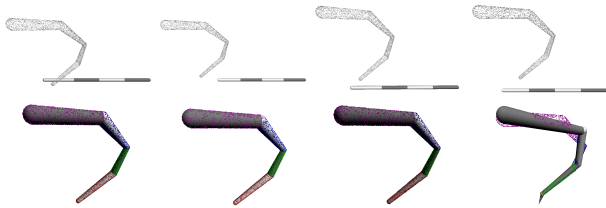


Fig. 17. Evaluation of FAKIR with respect to a bad initial anchor point position after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain. The last column shows that due to a bad initialization, the points (plotted in red) that are affected to the first bone do not bring enough information for the one-bone energy function to move the chain extremity. Then, not enough bones remain to approximate the whole point set.

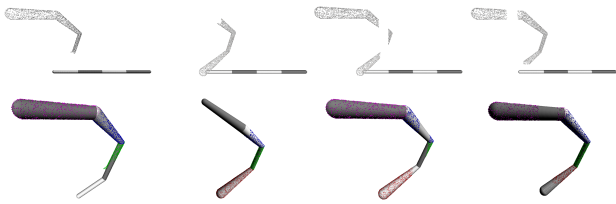


Fig. 18. Evaluation of the FAKIR algorithm with respect to missing data after 20 iterations. The first row shows the initial point set and the bottom row shows the registered bone chain.

- (5) Esquiline Venus: found in 1874 at the Horti Lamiani in Rome. Capitoline Museums, Rome, Italy
- (6) Old Fisherman Vatican-Louvre (or Dying seneca): found in Rome. Louvre Museum, Paris, France
- (7) Venus de Milo: Louvre Museum, Paris, France
- (8) Prince Paris: Ny Carlsberg Glyptotek, Copenhagen, Denmark.

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. Figure 21 shows obtained registrations on four statues. It also compares resulting models after skeleton pose change and skinning with our method or skinning with Dual Quaternions. Our skinning method clearly improves the quality of the modified model near bone joints. 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, and the skinning process yields a plausible result.

As far as the complexity of FAKIR is concerned, the computational bottleneck lies in the assignment and re-assignment of each point several times during the optimization process. This reassignment takes place during the updates of the one-bone and two-bones energy updates. The number of these updates is related to the geometry of the surface and not to the number of points sampled on that surface. 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 model, the first forward pass of FAKIR takes 5s and the average execution time of one pass of the FAKIR process takes less than 2s.

We compare FAKIR with Pinocchio [4] in Figure 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 the FAKIR process 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).

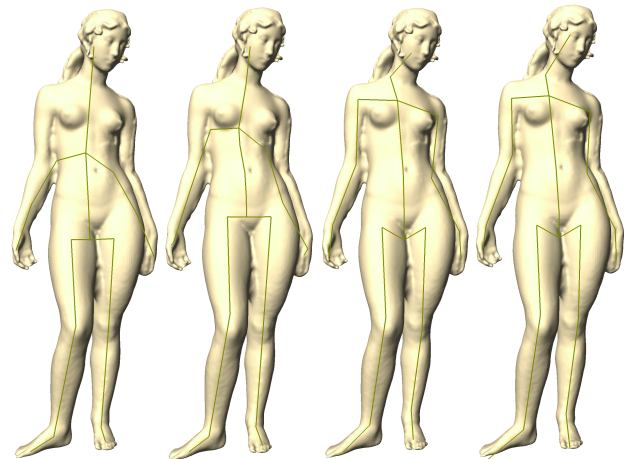


Fig. 19. Skeletons for the Aphrodite statue using the Pinocchio rigging method and comparison with our FAKIR algorithm. From left to right: Pinocchio with the Pinocchio-provided initial skeleton (17 bones); Pinocchio with our initial skeleton (22 bones); FAKIR with our initial skeleton after a single forward iteration; FAKIR with our initial skeleton in 10 iterations. Only the skeleton is displayed since the bone radii are not taken into account by Pinocchio.

5.3 Statue restoration by fragment combination

A direct application of statue pose and anatomy modification is statue restoration, by harmoniously combining parts belonging to different statues after bringing them to a common pose and anatomy. This type of statue restoration by part combinations was quite common in the 18th and 19th centuries¹. Performing this restoration virtually has two advantages: first, several hypotheses can be tested, second the pose and anatomy of the statues can be made similar before the combination, avoiding thus adding an oversized arm to an undersized body. Examples of statues with missing parts are shown in Figures 1 and 22. To complete a statue, we first register our anatomical model to the point set using FAKIR. In case there is no clue for the size a limb, such as for the missing arm, forearm and hand of the Esquiline Venus (Figure 22, first row), we use default human proportions. Then we choose a statue which is complementary to the incomplete statue and change its elementary anatomy and pose to match the ones of the incomplete statue. The last step is to integrate the selected parts into the first statue. We assume that the selected parts and the statue to complete slightly overlap, which it necessary to blend harmoniously statue parts.

The two point sets being brought to a common pose and anatomy, the combination area is defined as the area with two layers of points, one from each statue. Here we handle the case where there is only a single layer in each fragment above the sphere-mesh in the combination area. To prevent layers superposition artefacts, it is necessary to merge the information in these areas. In a nutshell, the merging consists in removing the lower layer in the overlap area, 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 Q_p be the subset of points on the two models that project on \tilde{p} , the projection of p on the sphere-mesh model, up to precision δ . 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 heightfield value is larger than $h_{max} - \delta$, where h_{max} is the maximum heightfield value for points in Q_p . Then we replace the heightfield value of p by a Gaussian-weighted average of the heightfield values in H_p . The resulting heightfield value of p is:

$$h(p) = \frac{1}{S} \sum_{q \in H_p} e^{-\frac{\|\tilde{p}-q\|^2}{2\delta^2}} h(q) \quad (8)$$

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.

In figure 1, we show the process of completing a statue which lacks arms and legs. Figure 22 shows the restoration of three 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 restoration method recovers plausible arms and legs in all three cases, leading to plausible restored statues.

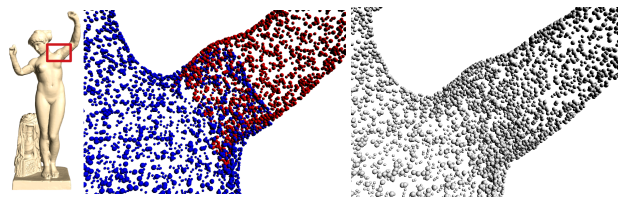


Fig. 20. In the left figure, blue points and red points come from different statues. The right figure shows the result after merging the two parts by taking the maximum followed by boundary smoothing.

5.4 Limitations

Our FAKIR algorithm works well to estimate the anatomical position of statues with or without clothes as long as the anatomical information remains visible. For example, in the *Dancer with Crotales* case (first row of Figure 21), the dimension and the pose of the legs is easy to infer although the legs are partially hidden by the dress. However if the clothes hide a large part of the statue, such as in the *Venus de Milo* case (third row in figure 22), FAKIR fails at recovering the anatomy and pose of the legs, as they are covered by the drape. For our point set model, a change in pose may reveal areas where there is no data. This occurs wherever two parts are stuck together in the initial pose. For example in Figure 21 (fourth row), there seems to be a veil between the right arm and the body (see also Figure 22, second row). Finally, combining parts of different statues for restoration can sometimes look unnatural because the materials and styles of the combined parts are different, style transfer would be necessary to alleviate this effect but this is a whole different research topic.

6 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. We also proposed a point set skinning method to modify the point set when the pose of a statue is changed. We compared our registration and skinning approaches with existing approaches to highlight their benefits. We also illustrated that this framework can be used to combine statue parts or add missing elements. While our method already gives good results, a further improvement would be to handle the case of a clothed statue which would involve modifying the FAKIR algorithm since anatomy parts may be hidden. Sometimes, the point cloud from the scan of a statue is incomplete, or some areas, occluded in the original statue, are revealed by the pose change, creating thus a hole in the point set. In that case an inpainting process is necessary and will be the topic for our future work. Finally, our current approach to combine parts remains very rudimentary and is similar to a union despite a slight smoothing. As a future work, we would like to design a more respectful approach to the geometry and style of different fragments.

Acknowledgments. This work was funded by the e-Roma project from the Agence Nationale de la Recherche (ANR-16-CE38-0009). The *Dancer with crotales* model is a point set of the Farman Dataset

¹<https://art.thewalters.org/detail/22879/torso-of-artemis-with-head-of-aphrodite/>

[10]. The other 7 statues data are sampled on meshes from the Sketchfab website: the *Dancing Faun* model is courtesy of Moshé Caine, the *Venus de Milo* model is courtesy of Sketchfab user "tux", and the other 5 statues models (*Aphrodite*, *Old Man Walking*, *Esquiline Venus*, *Old Fisherman* and *Prince Paris*) are courtesy of Geofrey Marchal.

REFERENCES

- [1] Dicko Ali-Hamadi, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, François Faure, Olivier Palombi, and Marie-Paule Cani. 2013. Anatomy Transfer. *ACM Trans. Graph.* 32, 6, Article 188 (Nov. 2013), 8 pages. <https://doi.org/10.1145/2508363.2508415>
- [2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: shape completion and animation of people. *ACM Trans. Graph.* 24 (2005), 408–416.
- [3] Andreas Aristidou and Joan Lasenby. 2011. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graph. Models* 73, 5 (Sept. 2011), 243–260. <https://doi.org/10.1016/j.gmod.2011.05.003>
- [4] Ilya Baran and Jovan Popović. 2007. Automatic Rigging and Animation of 3D Characters. *ACM Trans. Graph.* 26, 3, Article 72 (July 2007). <https://doi.org/10.1145/1276377.1276467>
- [5] Manuel Bastioni. 2000. Make human: Open source tool for making 3d characters. <http://www.makehumancommunity.org/>.
- [6] Jules Bloomenthal and Ken Shoemake. 1991. Convolution Surfaces. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*. ACM, New York, NY, USA, 251–256. <https://doi.org/10.1145/122718.122757>
- [7] Federica Bogo, Michael J. Black, Matthew Loper, and Javier Romero. 2015. Detailed Full-Body Reconstructions of Moving People from Monocular RGB-D Sequences. In *International Conference on Computer Vision (ICCV)*. 2300–2308.
- [8] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. 2011. Probabilistic Reasoning for Assembly-based 3D Modeling. *ACM Trans. Graph.* 30, 4, Article 35 (July 2011), 10 pages. <https://doi.org/10.1145/2010324.1964930>
- [9] Siddhartha Chaudhuri and Vladlen Koltun. 2010. Data-driven Suggestions for Creativity Support in 3D Modeling. *ACM Trans. Graph.* 29, 6, Article 183 (Dec. 2010), 10 pages. <https://doi.org/10.1145/1882261.1866205>
- [10] Julie Digne, Nicolas Audfray, Claire Lartigue, Charyar Mehdi-Souzani, and Jean-Michel Morel. 2011. Farman Institute 3D Point Sets - High Precision 3D Data Sets. *Image Processing On Line* 1 (2011). https://doi.org/10.5201/ipol.2011.dalmm_ps
- [11] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. 2004. Modeling by Example. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 652–663. <https://doi.org/10.1145/1015706.1015775>
- [12] Benjamin Gilles, Lionel Reveret, and Dinesh Pai. 2010. Creating and animating subject-specific anatomical models. *Computer Graphics Forum* 29, 8 (Dec. 2010), 2340–2351. <https://doi.org/10.1111/j.1467-8659.2010.01718.x>
- [13] Nils Hasler, Carsten Stoll, Bodo Rosenhahn, Thorsten Thormählen, and Hans-Peter Seidel. 2009. Estimating body shape of dressed humans. *Computers & Graphics* 33, 3 (2009), 211 – 216. <https://doi.org/10.1016/j.cag.2009.03.026> IEEE International Conference on Shape Modelling and Applications 2009.
- [14] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and Hans-Peter Seidel. 2009. A Statistical Model of Human Pose and Body Shape. *Computer Graphics Forum (Proc. EUROGRAPHICS), Blackwell*, 337-346 (2009) 28 (04 2009). <https://doi.org/10.1111/j.1467-8659.2009.01373.x>
- [15] David A. Hirshberg, Matthew Loper, Eric Rachlin, and Michael J. Black. 2012. Coregistration: Simultaneous Alignment and Modeling of Articulated 3D Shape. In *Computer Vision – ECCV 2012*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 242–255.
- [16] Alain Hus. 1955. Sculpture étrusque archaïque : le cavalier marin de la Villa Giulia. *Mélanges de l'école française de Rome* 67, 1 (1955), 69–126. <https://doi.org/10.3406/mefr.1955.7396>
- [17] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. 2012. A Probabilistic Model for Component-based Shape Synthesis. *ACM Trans. Graph.* 31, 4, Article 55 (July 2012), 11 pages. <https://doi.org/10.1145/2185520.2185551>
- [18] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O’Sullivan. 2007. Skinning with Dual Quaternions. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games (I3D '07)*. ACM, New York, NY, USA, 39–46. <https://doi.org/10.1145/1230100.1230107>
- [19] Ladislav Kavan and Jiří Žára. 2005. Spherical Blend Skinning: A Real-time Deformation of Articulated Models. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games (I3D '05)*. ACM, New York, NY, USA, 9–16. <https://doi.org/10.1145/1053427.1053429>
- [20] Binh Huy Le and J P Lewis. 2019. Direct Delta Mush Skinning and Variants. *ACM Trans. Graph.* 38, 4, Article 113 (July 2019), 13 pages. <https://doi.org/10.1145/3306346.3322982>
- [21] Dongwoon Lee, Michael Glueck, Azam Khan, Eugene Fiume, and Kenneth Jackson. 2012. Modeling and Simulation of Skeletal Muscle for Computer Graphics: A Survey. *Foundations and Trends® in Computer Graphics and Vision* 7 (01 2012), 229. <https://doi.org/10.1561/06000000036>
- [22] J. P. Lewis, Matt Cordner, and Nickson Fong. 2000. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-driven Deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 165–172. <https://doi.org/10.1145/344779.344862>
- [23] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.
- [24] Nadia Magnenat-Thalmann, Richard Laperrière, and Daniel Thalmann. 1988. Joint-dependent Local Deformations for Hand Animation and Object Grasping. In *Proceedings on Graphics Interface '88*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 26–33. <http://dl.acm.org/citation.cfm?id=102313.102317>
- [25] Alex Mohr, Luke Tokheim, and Michael Gleicher. 2003. Direct Manipulation of Interactive Character Skins. In *Proceedings of the 2003 Symposium on Interactive 3D Graphics (I3D '03)*. ACM, New York, NY, USA, 27–30. <https://doi.org/10.1145/641480.641488>
- [26] A. N. Newell. 1934. Gallo-Roman Religious Sculpture. *Greece and Rome* 3, 8 (1934), 74–84. <https://doi.org/10.1017/S0017383500002424>
- [27] Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus Gross, and Leonidas J. Guibas. 2005. Example-based 3D Scan Completion. In *Proceedings of the Third Eurographics Symposium on Geometry Processing (SGP '05)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, Article 23. <http://dl.acm.org/citation.cfm?id=1281920.1281925>
- [28] Edoardo Remelli, Anastasia Tkach, Andrea Tagliasacchi, and Mark Pauly. 2017. Low-Dimensionality Calibration Through Local Anisotropic Scaling for Robust Hand Model Personalization. In *The IEEE International Conference on Computer Vision (ICCV)*.
- [29] Andrea Tagliasacchi, Thomas Delame, Michela Spagnuolo, Nina Amenta, and Alexandru Telea. 2016. 3D Skeletons: A State-of-the-Art Report. *Computer Graphics Forum* 35 (05 2016). <https://doi.org/10.1111/cgf.12865>
- [30] Jean-Marc Thiery, Emilie Guy, and Tamy Boubekeur. 2013. Sphere-Meshes: Shape Approximation using Spherical Quadratic Error Metrics. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia 2013)* 32, 6 (2013), Art. No. 178.
- [31] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. 2016. Sphere-meshes for Real-time Hand Modeling and Tracking. *ACM Trans. Graph.* 35, 6, Article 222 (Nov. 2016), 11 pages. <https://doi.org/10.1145/2980179.2980226>
- [32] Rodolphe Vaillant, Loïc Barthe, Gaël Guennebaud, Marie-Paule Cani, Damien Rohmer, Brian Wyvill, Olivier Gourmel, and Mathias Paulin. 2013. Implicit skinning: real-time skin deformation with contact modeling. *ACM Trans. Graph.* 32, 4, Article 125 (July 2013), 12 pages. <https://doi.org/10.1145/2461912.2461960>
- [33] Li-Chun T. Wang and Chih-Cheng Chen. 1991. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Trans. Robotics and Automation* 7 (1991), 489–499.
- [34] Xiaohuan Corina Wang and Cary Phillips. 2002. Multi-weight Enveloping: Least-squares Approximation Techniques for Skin Animation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '02)*. ACM, New York, NY, USA, 129–138. <https://doi.org/10.1145/545261.545283>
- [35] Chao Zhang, Sergi Pujades, Michael J. Black, and Gerard Pons-Moll. 2017. Detailed, Accurate, Human Shape Estimation From Clothed 3D Scan Sequences. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Appendix A ONE-BONE DISTANCE COMPUTATION

In this appendix, we detail the projection of a point p on a bone B . 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 will 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 the following computations depend on an angle α defined in Fig. 23 and which can be expressed as

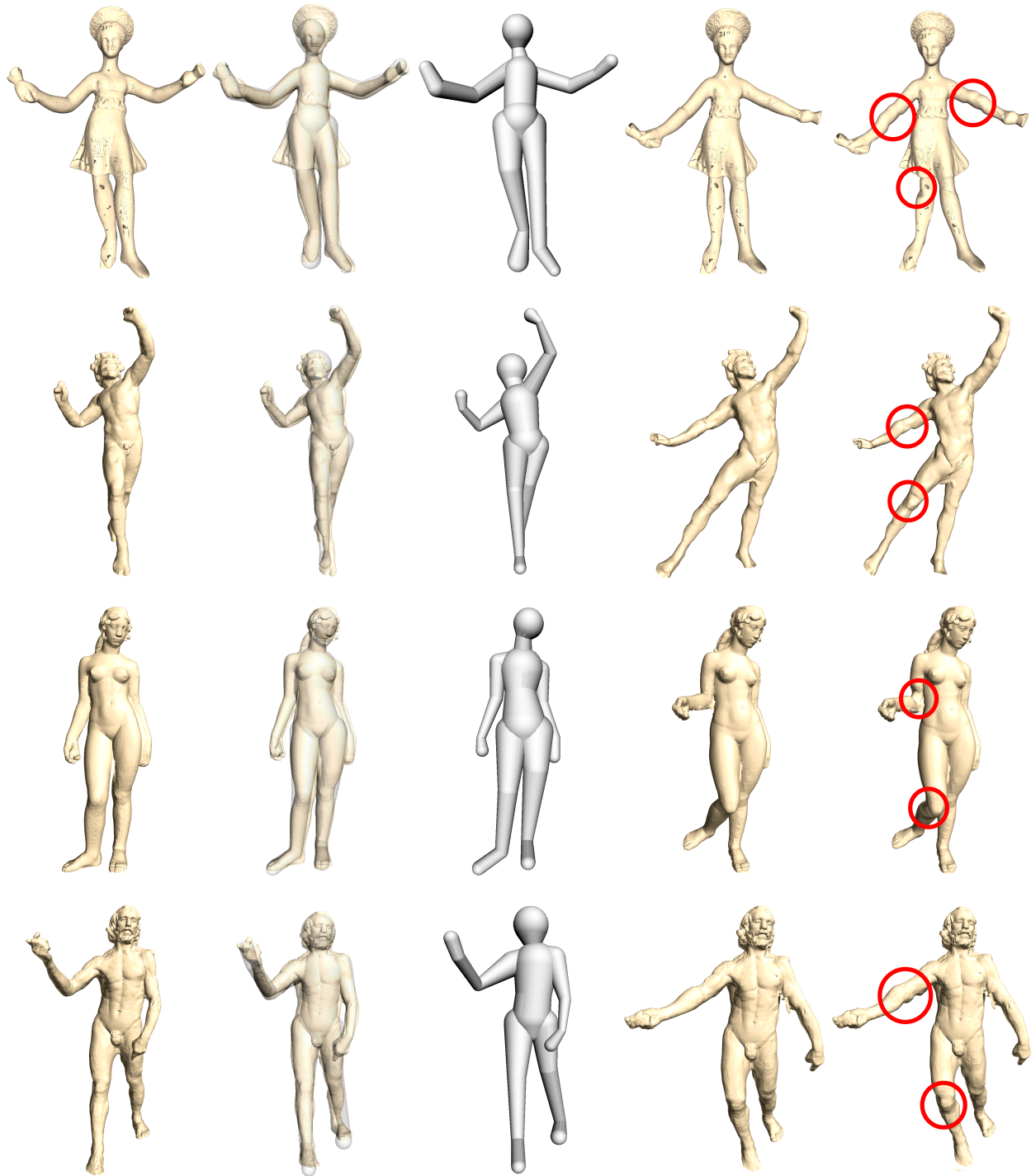


Fig. 21. Registration and pose changes of 4 statues: the Dancer with Crotales (first row), the Dancing Faun (second row), Aphrodite (third row) and the Old Man Walking (fourth row). First column: initial point set, second column: overlay of the registered model and the point cloud, third column: registered model, fourth column: final point set in a modified pose by our skinning method, fifth column: skinning result with Dual Quaternion. 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.

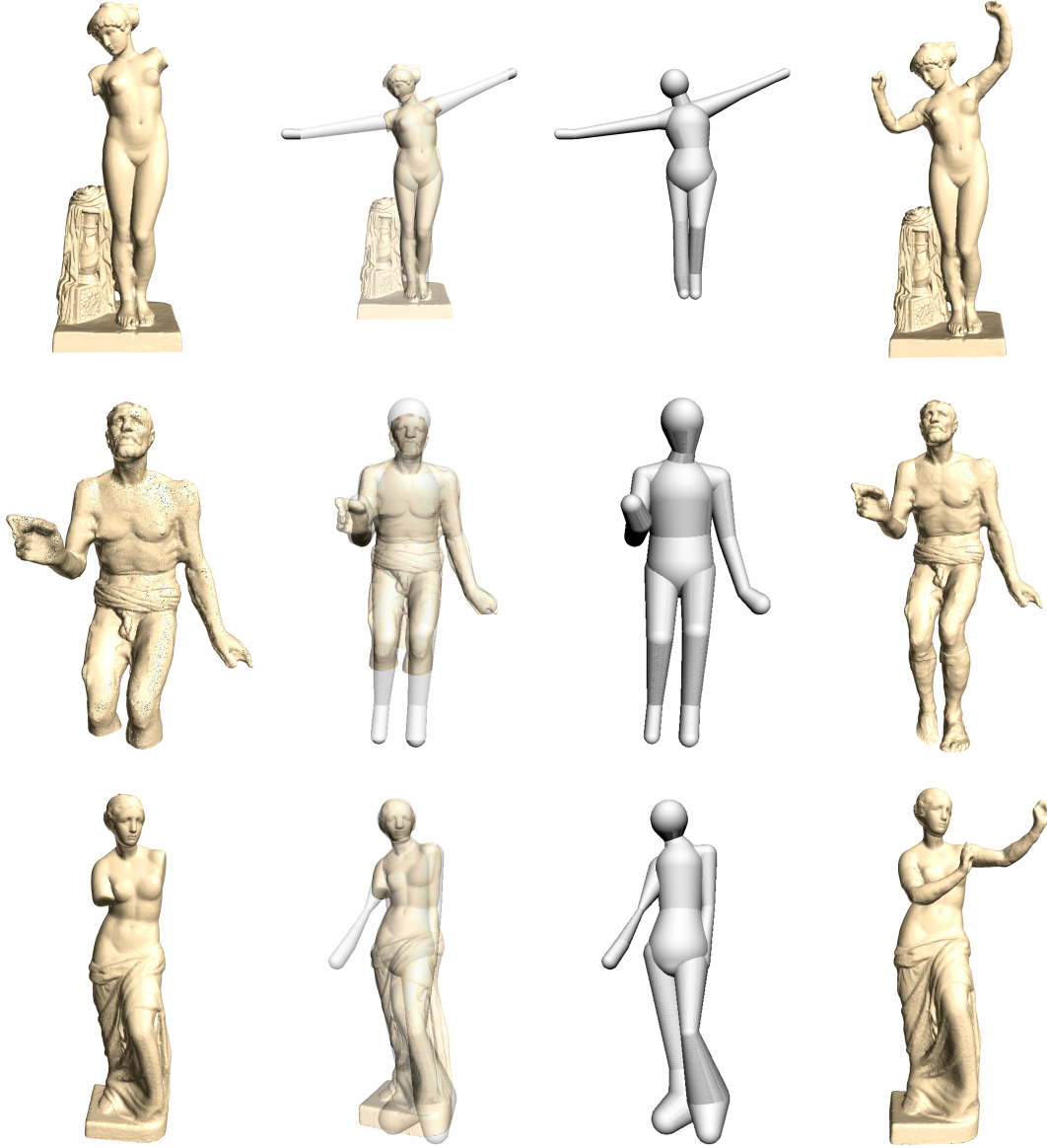


Fig. 22. Registration and restoration of 3 incomplete statues, Esquiline Venus (first row), Old Fisherman (second row) and Venus de Milo (third row). First column: initial point set, second column: overlay of the registered model and the point set, third column: registered model alone, fourth column: final restoration.

$\alpha = \arctan \frac{|r_2 - r_1|}{\sqrt{\|c_1 c_2\|^2 - (r_2 - r_1)^2}}$. Let us first compute p^* the projection of p on line $c_1 c_2$, 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 23. 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 intersection of $p_{-\alpha} p$ with the cone that is the farthest from p .

- $\tau_\alpha < 0$ (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 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 23, $\tilde{q}_{-\alpha}$ 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. In this situation, we compute the distances between point q and its projections on the two spheres and choose the closest projection point for \tilde{q} .

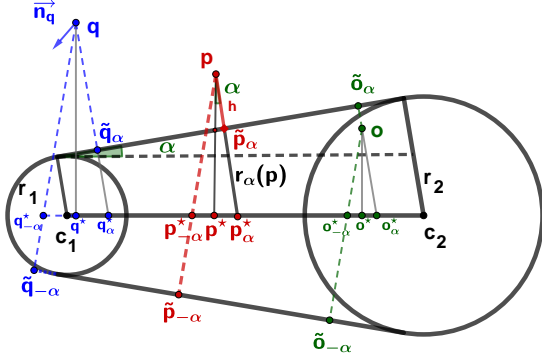


Fig. 23. Various projection cases. p has two possible projections \tilde{p}_α and $\tilde{p}_{-\alpha}$ depending on the orientation of the normal at p . Point p^\star is the projection of p on line c_1c_2 . If the normal at p is oriented upward $\tilde{p} = \tilde{p}_\alpha$. Otherwise, $\tilde{p} = \tilde{p}_{-\alpha}$. The same strategy is used to project points q and o .

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. $\tau_\alpha > 1$), $d(p) = \|c_1p\| - r_1$ (resp. $d(p) = \|c_2p\| - r_2$). If $0 \leq \tau \leq 1$:

$$d(p) = \begin{cases} \|pp_\alpha^\star\| - r_\alpha(p) & \text{if } n_{\tilde{p}} \cdot n_p > 0 \\ \|pp_{-\alpha}^\star\| + r_{-\alpha}(p) & \text{if } n_{\tilde{p}} \cdot n_p \leq 0 \end{cases} \quad (9)$$

Since the radius of the cone varies linearly along line c_1c_2 :

$$\begin{aligned} r_\alpha(p) &= \|p_\alpha^\star p_\alpha^\star\| = \tau_\alpha(p)r_1 + (1 - \tau_\alpha(p))r_2 \\ r_{-\alpha}(p) &= \|p_{-\alpha}^\star p_{-\alpha}^\star\| = \tau_{-\alpha}(p)r_1 + (1 - \tau_{-\alpha}(p))r_2 \end{aligned} \quad (10)$$

with: $\tau_\alpha(p) = \frac{c_1p_\alpha^\star \cdot c_1c_2}{\|c_1c_2\|^2}$ and $\tau_{-\alpha}(p) = \frac{c_1p_{-\alpha}^\star \cdot c_1c_2}{\|c_1c_2\|^2}$. Furthermore $\|pp_\alpha^\star\| = \|pp_{-\alpha}^\star\| = \|pp^\star\|/\cos \alpha$. Hence, for each bone, we first compute the α angle, then, for each point p , we compute its projection p^\star on c_1c_2 and the corresponding $\tau_\alpha(p)$ yielding $r_{\pm\alpha}(p)$ and $\tilde{p}_{\pm\alpha}$.

Appendix B OPTIMIZATION FOR ONE BONE

Let us assume that c_1 (Fig. 23) 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_1c_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 $\theta = (\theta_1, \theta_2), l$ and $r = (r_1, r_2)$. We have:

$$\tan \alpha = \frac{r_2 - r_1}{\sqrt{l^2 - (r_2 - r_1)^2}}, \cos \alpha = \frac{\sqrt{l^2 - (r_2 - r_1)^2}}{l}$$

$$\|c_1p\|^2 = x^2 + y^2 + z^2$$

$$\|c_1p^\star\| = x \cos \theta_2 \cos \theta_1 + y \sin \theta_2 + z \cos \theta_2 \sin \theta_1$$

$$\|p^\star p\|^2 = x^2 + y^2 + z^2 - (x \cos \theta_2 \cos \theta_1 + y \sin \theta_2 + z \cos \theta_2 \sin \theta_1)^2$$

$$\|p^\star p_\alpha^\star\| = \|p^\star p\| \tan \alpha$$

$$\|p_\alpha^\star p\| = \frac{\|p^\star p\|}{\cos \alpha}$$

$$\tau_{\pm\alpha}(p) = \frac{\|c_1p^\star\| \pm \|p^\star p_\alpha^\star\|}{l}$$

$$\|c_2p\|^2 = (x - l \cos \theta_2 \cos \theta_1)^2 + (y - l \sin \theta_2)^2 + (z - l \cos \theta_2 \sin \theta_1)^2$$

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

$$E(P, B(l, r), \theta) = \sum_{p \in P} d(p)^2 \quad (11)$$

The optimization is performed on three set of parameters in turn: angles θ , bone length l and bone radii r .

the optimization for bone B with respect to θ writes:

$$\hat{\theta} \equiv \underset{\theta}{\operatorname{argmin}} E(P, B(l, r), \theta) = \underset{\theta}{\operatorname{argmin}} \sum_{p \in P} d(p, \theta)^2 \quad (12)$$

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

$$\underset{\theta}{\operatorname{argmin}} E(P, B(l, r), \theta) \approx \underset{\delta\theta}{\operatorname{argmin}} E(P, B(l, r), \theta + \delta\theta) \quad (13)$$

which is computed by taking:

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

We finally get $\delta\theta$:

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

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 $g(\theta)$ is a column vector whose entries are $d(p, \theta)$ 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 $d(p) = \|pp_\alpha^\star\| - r_\alpha(p)$ with $r_\alpha(p) = \tau_\alpha r_1 + (1 - \tau_\alpha)r_2$, and $\tau_\alpha = \frac{\|c_1p_\alpha^\star\|}{l}$. Hence:

$$\frac{\partial d(p)}{\partial \theta_1} = \frac{1}{\cos \alpha} \frac{\partial \|p_\alpha^\star p\|}{\partial \theta_1} + (r_1 - r_2) \frac{1}{l} \frac{\partial \|c_1p^\star\|}{\partial \theta_1} + \tan \alpha \frac{\partial \|p^\star p\|}{\partial \theta_1} \quad (14)$$

$$\frac{\partial d(p)}{\partial \theta_2} = \frac{1}{\cos \alpha} \frac{\partial \|p_\alpha^\star p\|}{\partial \theta_2} + (r_1 - r_2) \frac{1}{l} \frac{\partial \|c_1p^\star\|}{\partial \theta_2} + \tan \alpha \frac{\partial \|p^\star p\|}{\partial \theta_2} \quad (15)$$

The full expression for the derivatives can be easily derived given the expressions for $\|p_\alpha^\star p\|$, $\|c_1p^\star\|$, $\|p^\star p\|$ above. The cases $\tau_\alpha(p) < 0$, $\tau_\alpha(p) > 1$ or $n_{\tilde{p}} \cdot n_p < 0$ can be computed similarly.

Appendix C OPTIMIZATION FOR A JOINT BETWEEN TWO CONSECUTIVE BONES

Let us consider the geometric optimization of the center of the joint between two bones by optimizing the two-bones energy with respect to the lengths l_k and l_{k+1} . Each length is optimized in turn, with a side-effect on the value of the other length. The two-bones energy can be expressed as a function of l_k :

$$E_{(k,k+1)}(l_k) = \sum_{p \in P_k} \|\tilde{p}_k - p\|^2 + \sum_{p \in P_{k+1}} \|\tilde{p}_{k+1} - p\|^2 \quad (16)$$

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

$$\underset{l_k}{\operatorname{argmin}} E_{(k,k+1)}(l_k) \approx \underset{\delta l}{\operatorname{argmin}} E_{(k,k+1)}(l_k + \delta l) \quad (17)$$

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

$$\delta l = - \frac{\sum_{p \in P_k} d_k \frac{\partial d_k}{\partial l_k} + \sum_{p \in P_{k+1}} d_{k+1} \frac{\partial d_{k+1}}{\partial l_k}}{\sum_{p \in P_k} \left(\frac{\partial d_k}{\partial l_k}\right)^2 + \sum_{p \in P_{k+1}} \left(\frac{\partial d_{k+1}}{\partial l_k}\right)^2} \quad (18)$$

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 6). 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 23 and appendix B, recall that $d_k = \|p - \tilde{p}_k\| = \|pp^\star\| - r_\alpha(p)$. Since when optimizing l_k the orthogonal projection on $c_k c_{k+1}$ does not change, $\|pp^\star\|$ remains the same. However both α and $r_\alpha(p)$ change. Since $r_\alpha(p) = \tau_\alpha(p)r_k + (1 - \tau_\alpha(p))r_{k+1}$ with $\tau_\alpha(p) = \frac{\|c_k p^\star\|}{l_k}$, we get:

$$\frac{\partial d_k}{\partial l_k} = - \frac{\|pp^\star\|}{\cos^2 \alpha} \frac{\partial \cos \alpha}{\partial l_k} - (r_k - r_{k+1}) \frac{\partial \tau_\alpha(p)}{\partial l_k} \quad (19)$$

Simple geometric considerations give $\cos \alpha = \sqrt{1 - \frac{(r_{k+1} - r_k)^2}{l_k^2}}$, $\tau_\alpha(p) = \frac{\|c_k p^\star\| + \|pp^\star\| \tan \alpha}{l_k}$ 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:

$$\frac{\partial l_{k+1}}{\partial l_k} = \frac{1}{\cos \alpha} \frac{\partial \|pp^\star\|}{\partial l_k} - \frac{\|pp^\star\|}{\cos^2 \alpha} \frac{\partial \cos \alpha}{\partial l_k} - (r_{k+1} - r_{k+2}) \frac{\partial \tau_\alpha}{\partial l_k} \quad (20)$$

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

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

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

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

Plugging all the derivatives in Equation 18 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 $c_k c_{k+2}$ are done in a similar manner.