Anatomy and pose estimation from point sets using FAKIR

Tong Fu, Raphaëlle Chaine, Julie Digne

Journées Françaises d'Informatique Graphique et de Réalité Virtuelle, Marseille, 12-15 novembre 2019

Anatomy and pose estimation from point sets using FAKIR

Tong FU[1] , Raphaëlle Chaine[1] et Julie Digne [1]

[1]LIRIS, CNRS, Université de Lyon

Résumé

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.

Mots clés : Skeleton registration, point set analysis, anatomy detection

1. Introduction

The progress of 3d scanning techniques gives the possibility to create digital replicas of artworks as well as to build and test restoration hypotheses for damaged artifacts. 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 focus on human statues with no or few garments which 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 estimate the elementary anatomy and the pose of a statue allowing in turn to change the statue's pose. To do so, we 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.

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.

This paper is a short version of [FCD19].

2. Related work

Anatomical Model. Designing anatomical models for human shapes has raised a lot of interests. 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 the human skeleton representation and model every single muscle to increase realism [LGK*12], we will focus here on skeletons, which are simple and efficient enough for our purpose. Among skeleton-based models, the sphere-mesh model [TGB13] has been introduced for representing mesh models by packing sphere into it and encoding its structure. This model is light and pliable and we will also rely on it.

Skeleton rigging. Once a skeleton model is chosen, the next problem for animation purposes is to position it inside an input mesh, which we call the process *rigging*. The Pinocchio algorithm [BP07] packs spheres into the mesh and considers their centers as the admissible joint positions. This pre-computation makes the skeleton pose estimation tractable. It is possible to infer, or track, a skeleton from a dynamic input data. 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 admis-

sible poses [TPT16, RTTP17]. Such methods require a dynamic scene and cannot apply to the static mesh rigging problem.

3. Anatomy and Pose estimation

3.1. Human model

Our work focuses on artworks representing human beings without or with only a few garments. Statues do not often follow the real human proportion beauty canons because of the artistic aesthetic perception. In this context, it is necessary to devise a human model with few constraints allowing to fit an unrealistic sculpture.

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 $B(l, \mathbf{r})$ corresponding to the envelope of the union of a set of spheres centered on a segment and with a linearly varying radius (Figure 1). The sphere-mesh model is controlled by the length $l = \|c_1c_2\|$ and the pair of sphere radii $\mathbf{r} = \{r_1, r_2\}$ where c_1, c_2 are two end sphere centers and r_1, r_2 are the associated radius respectively. The segment $[c_1c_2]$ is the medial axis of the bone. For each sphere center $c \in [c_1c_2]$, the radius of the sphere centered at c is $r(c) = (1-\tau)r_1 + \tau r_2$, with $\tau = \frac{\|c_1c\|}{\|c_1c_2\|}$. We denote by α the angle of the conic part of the bone, as illustrated on Figure 1. 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*.

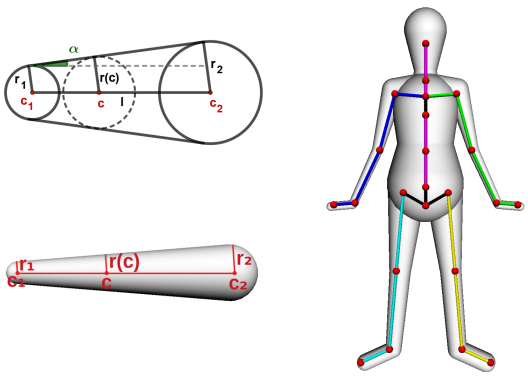

Figure 1: Left figure shows a 2D cross-section and a 3D sphere-mesh of a bone. Right figure shows our anatomical human model with its control skeleton.

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 1 : 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. Each bone is thus fully defined by its intrinsic parameters (length l and two radii $\mathbf{r}$) and by its extrinsic parameter $\boldsymbol{\theta}$ (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. FAKIR : Forward And bacKward Iterative Registration

Inspired by the FABRIK [AL11] and CCD [WC91] 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.

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. $\tilde{p_k}$ is a normal-constrained projection of point p on bone B_k by using the normal vector $\mathbf{n}_p$ to disambiguate the choice between several orthogonal projection possibilities.

Registration process for a chain of bones. We obtain a first approximation of the registration for each bone by the minimization of the one-bone energy using the Levenberg-Marquardt algorithm for each parameter.

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

In the case of the minimization with respect to the vector of rotation angles $\boldsymbol{\theta}_k$, we iteratively try to replace the current angles $\boldsymbol{\theta}_k$ with an update $\boldsymbol{\theta}_k + \delta\boldsymbol{\theta}_k$. At a minimum, $\nabla_{\delta\boldsymbol{\theta}_k} E_k(P_k, B_k(l_r, \boldsymbol{r}_k), \boldsymbol{\theta}_k + \delta\boldsymbol{\theta}_k) = \mathbf{0}$, and the value for $\delta\boldsymbol{\theta}_k$ follows. Our algorithm gradually rotates the current bone B_k with respect to its predecessor, updating P_k after each rotation, so that P_k

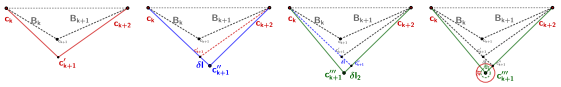

Figure 2: Pairwise Optimization. With fixed extremities c_k and c_{k+2}, the pair of bones B_k and B_{k+1} is first rotated around axis $c_k c_{k+2}$ in order to minimize the two-bones energy. Then the lengths of the bones B_k and B_{k+1} and their common radius r_{k+1} are optimized successively. After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

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}. We use this energy only for initialization since it might be inefficient to approximate the full length of a bone accurately.

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.

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

Figure 2 illustrates the procedure between two consecutive bones. We refer the reader to [FCD19] for the computation details.

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.

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.

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.

Initialization :
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 ;
for $k := 1$ **to** $K - 1$ **do**
  Consider the pair of bones B_k, B_{k+1} :
  Fix the position of the joint common to B_k and B_{k+1} ;
  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} ;
  Fix the positions of the 2 joints that B_k and B_{k+1} do not share, and free their common joint ;
  Compute the position and the radius of the common joint by using the two-bones energy function.
end for
Compute the length of the last bone and the radius of the last sphere.
Forward and Backward registration loop :
repeat
  Reverse the order of the bones in the chain ;
  for $k := 1$ **to** $K - 1$ **do**
    Consider the pair of bones B_k, B_{k+1} :
    Fix the positions of the 2 joints that B_k and B_{k+1} do not share ;
    Compute the position and the radius of the common joint by using the two-bones energy function.
  end for
  Compute the length of the last bone and the radius of the last sphere with the one-bone energy function.
until convergence

4. 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.

4.1. Experiments on synthetic data

We first test our algorithm on synthetic data to provide a quantitative evaluation of the FAKIR performances. The accuracy of the registration is evaluated

as the average distance between the point set and the model :

$$dist = \frac{1}{N_{points}} \sum_{p \in P} \|p - \tilde{p}\|. \quad (3)$$

For a point set of $5k$ points sampled on a sphere-mesh of a 4-bone chain in a specific pose 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 3 : 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 (second column in figure 4). The third column in figure 4 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 also rather robust to missing data thanks to the iterated forward and backward passes (fourth column in figure 4).

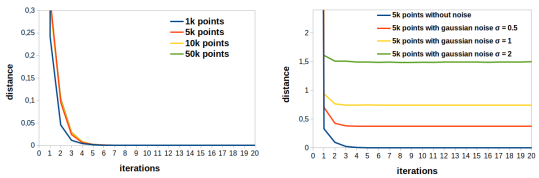

Figure 3: Evolution of the registration distance with the iterations for different number of points in the point set (left image) and different levels of noise (right image).

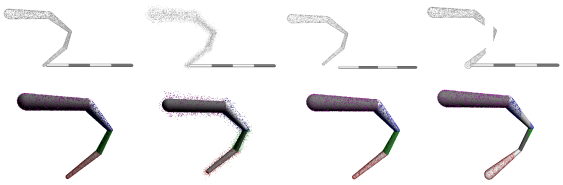

Figure 4: Evaluation of FAKIR in different situations after 20 iterations.The first row shows the initial point set and the bottom row shows the registered bone chain. From left to right : an ideal initial condition ; point set with a Gaussian noise : $\sigma = 2$; the anchor point is far from the point set ; point set with missing data. The total groundtruth model length is 140.

4.2. Skeleton registration results on statues

The registration algorithm performs well for statues depicting naked characters.Even with moderate clothing or a incomplete statue (*Wounded Amazon*) FAKIR recovers the pose of the statue. 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 $2s$.

Figure 5: Registration results of 2 statues : *Wounded Amazon (Rome, 150 A.D., Nye Carlsberg Glyptotek, Copenhagen, Denmark)* and *Age of Bronze (a bronze statue by the French sculptor Auguste Rodin (1840–1917))*. From left to right : initial point set, overlay of the registered model and the point cloud, registered model.

We compare FAKIR with Pinocchio [BP07] in Figure 6. 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).

5. 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. 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.

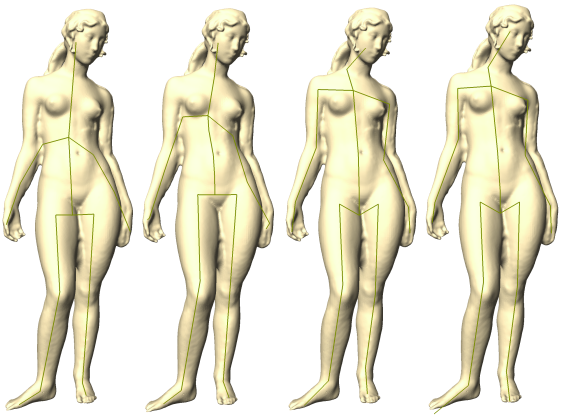

Figure 6: From left to right : Pinocchio with the Pinocchio-provided initial skeleton (17 bones) ; Pinocchio with our initial skeleton (22 bones) ; FAKIR with our initial skeleton after a single forward iteration ; FAKIR with our initial skeleton in 10 iterations. Only the skeleton is displayed since the bone radii are not taken into account by Pinocchio.

Acknowledgments This work was funded by the e-Roma project from the Agence Nationale de la Recherche (ANR-16-CE38-0009). The statues used in this paper are sampled on meshes from the Sketchfab website and are courtesy of of Geoffrey Marchal.

Références

[AL11] Aristidou A., Lasenby J. : FABRIK : A fast, iterative solver for the inverse kinematics problem. *Graph. Models. Vol. 73*, Num. 5 (septembre 2011), 243–260.

[Bas00] Bastioni M. : Make human : Open source tool for making 3d characters. μ`http://www.makehumancommunity.org`, 2000.

[BP07] Baran I., Popović J. : Automatic rigging and animation of 3d characters. *ACM Trans. Graph.. Vol. 26*, Num. 3 (juillet 2007).

[FCD19] Fu T., Chaine R., Digne J. : FAKIR : An algorithm for estimating the pose and elementary anatomy of archaeological statues. *arXiv e-prints* (Jul 2019), arXiv :1907.11721.

[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. Vol. 7* (01 2012), 229.

[RTTP17] Remelli E., Tkach A., Tagliasacchi A., Pauly M. : Low-dimensionality calibration through local anisotropic scaling for robust hand model personalization. In *The IEEE International Conference on Computer Vision (ICCV)* (Oct 2017).

[TGB13] Thiery J.-M., Guy E., Boubekeur T. : Sphere-meshes : Shape approximation using spherical quadric error metrics. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia 2013). Vol. 32*, Num. 6 (2013), Art. No. 178.

[TPT16] Tkach A., Pauly M., Tagliasacchi A. : Sphere-meshes for real-time hand modeling and tracking. *ACM Trans. Graph.. Vol. 35*, Num. 6 (novembre 2016), 222 :1–222 :11.

[WC91] Wang L.-C. T., Chen C.-C. : A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Trans. Robotics and Automation. Vol. 7* (1991), 489–499.