Baseline Skinning for Point Sets of Articulated Bodies

Tong Fu, Raphaëlle Chaine, Julie Digne

Université de Lyon, UCBL, CNRS, France

Abstract

General skinning techniques aim to deform the surface of an articulated model following the pose change of a skeleton. Their rapidity makes them ideal tools for real-time animation purposes. However, popular skinning algorithms are simple, but they tend to generate undesirable geometric artefacts. In our work, we consider skeletons given in the form of sphere-mesh models controlling both the pose and morphology of the shape that is either described as a mesh or a raw point set. We propose a novel skinning method that encodes the point set details above a bundle of baselines covering the sphere-mesh. In particular, we propose a geometrical model of the baseline and detail direction evolution during bone twisting and joints bending rotations. Our approach works directly on point sets and thus preserves the accuracy of the initial sampling. It further avoids computing a weight per point or a costly explicit muscle modelling step. We evaluate our method on several articulated body point sets, showing that it creates fewer artefacts than classical methods.

Keywords: point set skinning, shape deformation, sphere-mesh model.

1. Introduction

The main idea of skinning is to deform a surface following an underlying skeletal animation. Each surface point is influenced by one or more skeleton bones, and weights describe the influence of relevant bones on each point (usually a mesh vertex). Skinning techniques are widely used in computer animation or shape modeling to change a character’s pose and shape.

Many widely used geometric skinning methods, such as Linear Blend Skinning \([1]\) or Dual Quaternion skinning \([2]\), suffer from volume collapse problems and need a careful design of per-vertex weights. Example-based methods \([3]\) and physics-based methods do not suffer from such artefacts. However, they are computationally expensive and require additional input data.

Our goal is to provide a point set skinning technique that preserves the geometric detail and keeps the initial sampling accuracy. We assume that a skeleton, in the form of a sphere-mesh model, controls the pose and the dimensions of the overall shape of the point set. Our approach encodes the shape details as a vector field above a set of baselines covering the sphere-mesh model’s surface. We propose to associate a base point on the articulated sphere-mesh model to each of the original shape points. Each base-point is located on a baseline over the sphere-mesh model, and the position of a base-point after deformation follows its baseline’s motion. We then deduce the new position of an input point from its corresponding base-point’s new position by adapting its initial detail field value. Our approach’s originality is not to require a mesh with fixed connectivity whose triangles quality may be altered by deformations related to poses and anatomy changes, possibly creating triangle slivers and self-intersections. Meanwhile, our point set skinning process does not need to determine and compute weights.

Applications of our method include virtual artwork restoration. In virtual archaeology, one way of restoring statues is to combine parts from different statues after bringing them to a common pose and morphology and require hence artefact-free skinning results while avoiding the explicit simulation of muscles that may not respect the artistic style. Unlike an animation of 3D characters, skinning for virtual restoration is more sensitive to preserving the initial surface details. Besides, it requires not only rigid transformations but also affine transformations such as scaling. In this context, our baseline skinning method gives relevant results and preserves the details of the input shape.

To summarize, our main contributions are:

- A skinning method working directly on a point set.
- An approach to encode a point set above a sphere-mesh using a specific detail direction field.
- A full geometric model of the evolution of a detail field following a sphere-mesh model deformation.
- A skinning method able to handle multi-layered details deformation.
- A method that can handle articulated bodies such as human shapes as well as imaginary creatures.

2. Related work

The most famous and simple skinning method is Linear Blend Skinning (LBS for short) \([1]\). A vertex on a mesh surface is transformed by a linearly weighted combination of the motions related to the moving bones it is attached to. Despite well-known limitations (the candy wrapper effect and the elbow collapse effect), Linear Blend skinning is still the standard skinning method for real-time animation purposes. Many skinning approaches base on the principle of LBS and improve its limitations, such as Pose Space Deformation \([4]\), Log-matrix blending \([5, 6]\), Optimized centers of rotation \([7]\), Multi-Weight Enveloping \([8]\), spherical Skinning \([9]\) and Dual Quaternions Skinning \([2]\), Jacobson and Sorkine \([10]\) proposed to add extra endpoint weights to the LBS formulation so that it can handle properly stretch and twist. The endpoint weight of one vertex for one bone varies from 0 to 1, which indicates the original vertex’s position relative to the bone’s endpoints. The twist rotation’s angle of this vertex on the
bone is a linear combination of the twisting angle of the two end-points. The bend rotations are carried out afterwards on the positions resulting from the twist. Kavan and Sorkine [11] then proposed a joint-based deformor that differentiates bend and twist motions at each joint. The deformor applies a spherical interpolation of the twist (as DQS) and then a linear interpolation of the bend (as LBS). The weights for bend and twist are different which are precomputed separately as their energy-minimizing weights for linear blend skinning. Fu et al. [12] further improved the result by handling bend rotations anisotropically. But there are still some artefacts at joints such as a bulge effect. Furthermore, setting the suitable weights is an essential question for these skinning methods: while the profile of the weights is generally sketched by graphic designers [3], there exist automatic weighting techniques that, for example, use heat diffusion [13,14,15], geodesic voxel binding [16] or bounded biharmonic weights [17]. Applying suitable weights on the transformation matrices of bones gives a smooth transition at bones joints.

As defined above, Skinning processes allow deforming the skin following an underlying skeletal animation. In most skinning methods, the skin is a 3D mesh whose vertices are attached to a set of bones using different weights, and the skeleton is a tree whose nodes represent joints of the skeleton and edges represent bones. In this work, we propose to consider the skin of the model as a set of detail points, encoded as a set of displacements on top of a sphere-mesh articulated model and transferred back on this model after pose or morphology changes. The originality of our approach is that the direction in which the detail is encoded does not necessarily correspond to the normal, especially in concave areas of the sphere-mesh. Furthermore, this detail direction may evolve after a pose change. The detail direction field is hence not a simple heightfield [35].

Going in a different direction, physics-based methods simulate the growth of skeletal muscles and fat tissues using a biological model. Hamadi et al. [21] transfers the volume delimited by a mesh to the interior of another mesh by minimizing some harmonic energy. To do so, a deformation field is computed between the two meshes, based on the nearest point matching which is updated at each iteration [22]. However, the process is not fully automatic. Recent researches [23,24,25] achieve the desired deformation of the human body or face by direct control of each muscle. These methods are computationally expensive and need to define the physical model from physiological data. They are not suitable for artistic shapes since the morphology, and the representation of muscles in artwork may differ from a biological model.

Taking a different perspective on the problem, Implicit skinning [26] uses an additional implicit formulation of the surface that better supports pose changes and re-projects skinned vertices on the implicit model after each pose change. In this paper, we also use a proxy model, but it is explicit. Volume preserving skinning methods [27,28,29] correct for volume changes through the generation of extra bulges and wrinkles. They use vector fields induced by skeletal motion to describe the skin deformation. Some skinning methods involve a rough modeling of muscles combined with implicit skinning [30].

Example-based methods [31,32,33,34] produce more realistic results but require extra training data. They have a limitation for a given range of deformations. In addition, these methods can only be as good as their training dataset is, and for unrealistic characters (imaginary creatures or characters with unrealistic body proportions), relevant datasets are nontrivial to build.

3. Method

Our method uses a sphere-mesh model [14] as a skeleton guiding the shape deformation. A volumetric sphere-mesh is a union of spheres centered at a simplicial complex, with linearly varying radius between vertices. In our approach, we only consider 1D bones, which means that each bone \( B(l, r) \) corresponds to the union of a set of spheres centered on a segment with a linearly varying radius (Figure 2). The bone model is controlled by the length \( l = ||C_l C_2|| \) and the pair of sphere radii \( r = \{r_1, r_2\} \) where \( C_1, C_2 \) are the two end sphere centers and \( r_1, r_2 \) are the associated radius respectively. The segment \( C_1 C_2 \) is the medial axis of the bone. For each sphere center \( C \in C_1 C_2 \), the radius of the sphere centered at \( C \) is \( r(C) = (1 - \rho)r_1 + \rho r_2 \), with \( \rho = \frac{||C - C_l||}{||C_l C_2||} \). We denote by \( \alpha \) the angle of the conic of the bone, as illustrated on Figure 2. A chain of bones is a sequence of bones with each bone sharing a common end sphere with the next bone. The skeleton that we use is a tree composed of several chains of bones connected at some junctions spheres. We further assume that this sphere-mesh model is registered to the input point set,
The model is calibrated to match the input shape, and we know which point is associated to which bone. In practice, to get this initial registration, we use the FAKIR algorithm [37].

Baseline definition. Given a chain of bones, a baseline is a continuous curve defined on the surface of the sphere-mesh. Before a pose change, it is composed initially of a constrained sequence of segments and circular arcs. The segments are carried by the conical parts of the bones (generators), and the circular arcs are carried by the sphere caps at the articulation between two bones. The constraints on the circles and segments composing the baseline are as follows:

- There is one segment per bone.
- The segments belonging to two consecutive bones are constrained to belong to the same plane containing the apexes of the two cones. If they do not share a common endpoint, these two segments are connected in a $C^1$ manner by a circular arc also belonging to the plane of the two segments.

Figure 3 illustrates two portions of baselines on a two bones chain. The one in green contains a circular arc, while the one in blue illustrates the case where two segments directly share a common endpoint.

Building a baseline. Starting from a point on a cone part of the sphere-mesh, we consider the cone generatrix that passes through that point: the first segment is the part of the generatrix corresponding to the sphere-mesh surface restricted to that cone. Then, the rest of the baseline is built using the following principle: Given a baseline segment on one bone $B_k$, the corresponding baseline segment on subsequent bone $B_{k+1}$ is obtained by intersecting the sphere-mesh surface restricted to $B_{k+1}$ with the plane containing the segment on $B_k$ and the cone apex $A_{k+1}$. By construction, this plane also contains the cone apex $A_k$. There are two connected components in the intersection of the plane with the two truncated cones and their connecting spherical cap. We choose the segment of $B_{k+1}$ that lies in the same connected component of our initial segment on $B_k$. If these two segments do not have a common endpoint, they are connected in a continuously differentiable ($C^1$) way by a circular arc. The latter is a circular portion of the intersection between the plane of the two segments and the spherical cap shared by the two bones. The same construction process is valid for extending a baseline segment of a bone onto the preceding bone.

In degenerate cases where one of the bones is, in fact, a cylinder, a baseline segment on that bone is a segment parallel to the cylinder axis, and the corresponding segment in an adjacent bone is included in the plane defined by the segment on the cylinder and the apex of the other cone. If both bones are cylinders, the plane containing the baseline segments on those bones is parallel to the axis of both cylinders.

If one bone corresponds to a free extremity of the sphere-mesh, a baseline segment on that bone is extended by a circular arc joining the endpoint to the furthest point of the sphere cap that is aligned on the bone axis (see Figure 3). Bundle of baselines. It is essential to notice that the baselines built for a chain of bones cannot cross each other. Hence, the surface of each chain of bones is covered by a bundle of non-intersecting baselines.

Bundle of baselines at junctions. A sphere-mesh is usually composed of several chains of bones, which join at a common sphere end. It is possible to extend the construction of the bundle of baselines by subdividing the sphere into several portions in the manner of a Voronoi diagram. Each cell comprises the points that are closer to a tangential cone portion than to another tangential cone portion. On the sphere, the median between two cones is a circular arc in the plane that holds the intersection of the two cones. It is indeed a plane because the cones are tangent to a common sphere. At the junction, baselines connect coplanar segments of two cones if the plane holding these two segments also contains a common point to the two cones on the union of all the cones truncated with each other.

Detail direction field. Armed with this baseline definition, we now define how a continuous vector field of detail directions can be defined over the sphere-mesh using the system of baselines.
3.3. Projection of a point on the sphere-mesh using the detail direction field

Given a point of the input shape, we propose to project it on the sphere-mesh using detail directions induced by baselines in-}

stead of using the orthogonal projection. Hence, we can obtain a more natural and continuous way of modelling the details in the concave parts of the sphere-mesh. More precisely, given a point \( p \), we are confronted with the inverse problem of determining a base-point \( b_p \) of the sphere-mesh such that \( p \) can be encoded as a displacement in the detail direction above \( b_p \). Fortunately, this inverse problem is made easy because, at a joint and its two incident bones, the segments and circular arcs of a baseline are coplanar, and by the fact that detail directions are also coplanar above each baseline segment.

**Baseline determination.** Given \( p \), we first have to identify the baseline portion that contains base-point \( b_p \). Assuming that the local variation of a detail point after a skeletal movement is only influenced by its closest bone, the predecessor bone and the successor bone, the portion of baseline associated to \( p \) only needs to be computed over at most three bones. First, let us first deal with the easy case where point \( p \) projects orthogonally on a spherical cap of the closest bone. In this case, the base-point \( b_p \) coincides with the orthogonal projection of \( p \) onto the sphere and the detail direction \( \vec{n}_B \) corresponds to the normal to the sphere at this point. Hence the base-point is trivial to find, and the local portion of the baseline is built accordingly on the two adjacent bones. In the nontrivial case where the orthogonal projection of \( p \) lies on a cone part of the closest bone, the base-point must be determined on the baseline associated with \( p \). Our method processes the bones by pairs: by first considering the portion of baseline along \( p \)'s closest bone and its successor bone to deal with one joint and the portion of baseline on the predecessor bone and the closest bone with the other joint. Hence, the following explanations will be illustrated on a sphere-mesh corresponding to a single pair of bones. We refer to Figure 5 for the following notations. The example sphere-mesh model is composed of a sphere joint \((C_{k+1}, r_{k+1})\) and its two adjacent bones \(B_k\) and \(B_{k+1}\). Let \(A_k\) and \(A_{k+1}\) denote the apexes of the cones underlying \(B_k\) and \(B_{k+1}\).

\(\hat{p}\) is the orthogonal projection of \( p \) on its closest bone \(B_k\) and base-point \( b_p \) is such that \( p = b_p + h\vec{n}_B \), with \(\vec{n}_B\) the direction field at \( b_p \), and \( h \) the corresponding detail amplitude.

Those directions will be used to encode the detail points above the sphere-mesh.

First, along a circular arc, the direction of detail always coincides with the normal direction. The segment case is slightly more demanding: we first define the detail directions at the endpoints and then linearly interpolate along the segment.

- If the endpoint corresponds to the connection of the segment and a circular arc, the detail direction corresponds to the endpoint’s normal, which is well defined because of the \(C^1\) junction property.
- If the endpoint corresponds to the intersection of two cones tangent to a common sphere, the detail direction corresponds to the outward direction from the sphere’s center to the intersection point.

Figure 5 illustrates the profile of the detail direction field above a baseline. Note that along a baseline segment, the direction field vectors are coplanar by construction. The detail direction field along a segment is entirely contained in the plane defined by the segment and the axis of the bone it belongs to. In the next section, we use this property to define a non-orthogonal projection of a point onto the sphere-mesh. Note that the plane including details changes from one segment of the baseline to another or from one segment of the baseline to a circular arc.

3.3. Projection of a point on the sphere-mesh using the detail direction field

Given a point of the input shape, we propose to project it on the sphere-mesh using detail directions induced by baselines in-
to be distinguished from the base-point \( b_p \). We consider the plane containing line \( A_2A_k+1 \) and \( \tilde{p} \), and take its intersection with bones \( B_k \) and \( B_{k+1} \) and their common joint but excluding the spherical caps at the other end of the two bones. If plane \( B_2A_{k+1} \) is not tangent to sphere \( (C_k+1, r_k+1) \), this yields two possible pieces of baseline curve, instead of a single one (blue and green curves in Figure 5). Each piece is composed of two segments ((UV and XY in green or US_1 and S_2 Y) in blue in Figure 5) that are possibly joined by a circular arc (VX in green in Figure 5). Note that, both pieces of baseline may possibly enclose a circular arc. Out of these two possible baselines, we select the one that contains \( \tilde{p} \). Hence, in Figure 5 the blue curve is the baseline piece associated with point \( p \).

Two exceptional cases must be considered. If one of the adjacent cones is a cylinder (both its end-sphere radii are equal), we consider instead the plane parallel to the cylinder axis, containing the apex of the remaining cone and point \( \tilde{p} \). The second case is when both cones degenerate to cylinders, in which case we consider the plane parallel to both cylinders axis and containing \( \tilde{p} \).

**Anchor points of a baseline.** If there is a circular arc in the baseline portion associated with point \( p \) on the joint sphere \( (C_k+1, r_k+1) \), we denote by \( S_{k+1} \) the intersection point of the circular arc and the plane enclosing the intersection of the two cones. This plane is called the separator plane of the two bones. \( S_{k+1} \) is different from the midpoint of the arc \( VX \). If there is no circular arc, \( S_{k+1} \) is the intersection of the two extended segments \( UV \) and \( XY \) of the underlying cones, where \( V \) and \( X \) both belong to the sphere joint \( (C_k+1, r_k+1) \) in Figure 5. Here again \( S_{k+1} \) belongs to the separator plane of \( B_k \) and \( B_{k+1} \). \( S_{k+1} \) is the anchor point of the baseline on joint \((C_k+1, r_{k+1})\).

**Base-point determination.** The portion of baseline associated to one point \( p \) on its closest bone and its two adjacent bones is composed of three segments that are possibly joined by one or two circular arcs. In the example of Figure 5 the piece of baseline contains segments \( MS_k \), \( S_kS_{k+1} \) and \( S_{k+1}Y \), where \( S_k \) and \( S_{k+1} \) are the intersections of the extended segments \( MN \), \( UV \) and \( XY \).

If \( \tilde{p} \) lies on a circular arc of sphere \( (C_k, r_k) \), the detail direction is given by the unit vector pointing from \( C_k \) towards \( \tilde{p} \), and \( \tilde{p} \) is the base-point of \( p \). If \( \tilde{p} \) lies on a segment part supported by a bone \( B_k \), the detail direction encoding \( p \) is enclosed in the plane containing \( B_k \)'s axis and \( \tilde{p} \). To determine the corresponding base-point we use the fact that detail directions are obtained by coplanar linear interpolation between the detail directions at the segment endpoints. If the segment is adjacent to two circular arcs on the baseline curve, the detail direction locally corresponds to the normal to the cone and the base point \( b_p \) still coincides with \( \tilde{p} \). However, if a segment on \( B_k \) is adjacent to another segment on \( B_{k+1} \) (resp. \( B_{k-1} \)) the direction of detail at their common point \( S_{k+1} \) (resp. \( S_k \)) is given by the unit vector pointing from the center \( C_{k+1} \) (resp. \( C_k \)) of their common joint sphere toward \( S_{k+1} \) (resp. \( S_k \)). This case is illustrated in Figure 5. Introducing an additional point \( I \), lying at the intersection of line \( (C_k, \tilde{n}_l) \) and \( (C_{k+1}, \tilde{n}_r) \), \( b_p \) is found as the intersection of lines \( pI \) and \( UV \), and the detail direction is \( \tilde{n}_l = \frac{b_p \tilde{p}}{b_pI \tilde{p}} \), which corresponds to a linear interpolation along the segment between \( \tilde{n}_l \) and \( \tilde{n}_r \).

**3.4. Encoding a point set above a sphere-mesh**

Each point is projected onto the sphere-mesh to determine its base-point and its local amplitude of detail in its detail direction. The projection process is independent for each point and can therefore be parallelized. The input point set is then described by a set of sphere-mesh base-points with a height as illustrated in Figure 6. There can be several occurrences of a same base-point when the starting shape is multi-layered or has plumb lines.

![Figure 6: Illustration of a point set (in red) encoded as a detail set over a baseline.](image)

The base-points are shown in green and in yellow. The yellow base-points have more than one detail point over them. The detail may be negative when it is oriented inward.

**3.5. Baseline and detail direction field after pose change**

After skeletal movement, segments composing baselines will be modified differently depending on whether a bone is twisted or a joint is folded or unfolded. In our deformation approach, the segments are transformed into curves on the conic part of the bones. The deformed baselines will be composed of pieces of these curves such that they can still be connected by circular arcs as in the original baselines.

**Twist.** When a bone \( B_k \) is twisted around its axis with \( B_{k-1} \) remaining fixed, the points belonging to a baseline segment of \( B_k \) will be deformed by rotations around the cone axis. The angle of rotation of a point \( \tilde{p} \) on the segment depends on its normalized distance \( d \in [0, 1] \) to the endpoint closer to \( B_{k-1} \), in a similar way of the approach proposed in [10][12]. Normalized distances are measured between twisting \( B_k \). After twisting, the segment deforms into a curve depending on the profile of the angle of rotation as an increasing function of \( d \). The circular arcs remain unchanged. Figure 7 illustrates the deformation of a segment after twisting \( B_k \) with an angle \( \tau_{max} \). The angle of rotation applied along the segment is computed by using a cubic function: \( \tau(d) = -2\tau_{max}d^3 + 3\tau_{max}d^2 - d, d \in [0, 1] \). But any other profile provided by an artist or resulting from learning would be possible. The advantage of the cubic profile is that the tangent to the curve at the endpoints of the segment is preserved by twisting.

**Bend.** When a joint between two consecutive bones \( B_k \) and \( B_{k+1} \) is bent, the cone of \( B_{k+1} \) is rotated around the axis of the joint, and \( B_k \) remains fixed. Given a baseline, the rotated segment \( XY \) of \( B_{k+1} \) is no longer coplanar with \( UV \) on \( B_k \). Initially, the two segments were in the same plane of the sheaf passing through \( A_kA_{k+1} \). After bend rotation, each segment is in a different plane of the sheaf as illustrated in Figure 8. The planarity loss is also illustrated on Figure 9. Therefore we propose to deform segments on the surface of cones so that they can be connected again by a circular arc defined by a plane of the sheaf. Let us first explain where the new endpoints of these curves \( V \) and \( X' \) are positioned on the sphere.

In the following descriptions, we usually illustrate the case where \( UV \) and \( XY \) do not intersect initially and are connected
by a circular arc in the initial baseline. However, the case where initial segments $UV$ and $XY$ intersect is treated in the same way, by performing the deformation on the extended segments $UV'$ and $XY'$.

By definition, $V$ and $X$ belong to the circles that connect their respective cones to the common sphere. Let us call these circles the circles of tangency (see Figure 8). After rotation, we propose to move the endpoints $V$ and $X$ on their respective circles of tangency so that they are brought in a common plane of the sheaf containing $A_kA_{k+1}$ and the displacements of $V$ and $X$ on their respective circles are similar in a sense that takes the position and the radius of the circles of tangency into account. Figure 10 illustrates the conjoint determination of the coupled target positions $V'$ and $X'$ for endpoints of segments being deformed. The rotated segment $XY$ can be completed into a $C^1$ continuous planar baseline proposition using circular arc $XV_1$ and segment $V_1U_1$ (in blue) in the plane passing through $X, A_k$ and $A_{k+1}$. Likewise, segment $Y_1X_1$ and circular arc $X_1V$ complete the segment $UV$ into a $C^1$ continuous planar baseline proposition (in yellow) within a plane passing through $V, A_k$ and $A_{k+1}$. $V$ and $V_1$ (resp. $X$ and $X_1$) are on the circle of tangency which connects the red cone (resp. blue cone) to the common sphere. The target endpoint $V'$ and $X'$ must be located on the circular arcs $V_1V$ and $X_1X$ respectively.

In the corresponding subset of the sheaf of planes passing through $A_kA_{k+1}$, we are seeking the relevant intermediate plane defining the target endpoints $V'$, $X'$ that balances the distances of $V_1V'$ and $VV''$ but also $X_1X'$ and $XX'$. For this we cannot work directly on the circle of tangency of $B_1$, nor on the circle of tangency of $B_{k+1}$, but on a pivot circle at the intersection between the separator plane of the two cones and the joint sphere. This pivot circle is illustrated in purple in Figure 10. Let $E_{XY}$ (resp. $E_{UV}$) be the intersection of the baseline arc $V_1X$ (resp. $X_1V$) and the pivot circle. $E_m$ denotes the middle point of the circular arc $E_{XY}E_{UV}$ on the pivot circle. The plane of the sheet passing through $E_m$ is the intermediate plane. $V'$ and $X'$ are defined by the intersections of that plane with the tangential arcs $VV_1$ and $X_1X$ respectively.

In Figure 10, we illustrate how to find the relevant intermediate plane of the sheaf between a baseline passing in the convex area of the joint and a baseline passing in the concave area. Our approach works for all other situations that we illustrate in Figure 11. This Figure also illustrates that a baseline initially positioned in the convex part of a joint may evolve in the concave part after a bend, and vice versa.

The target position $V'$ (resp. $X'$) corresponds to a target rotation angle $\theta_k'$ (resp. $\theta_{k+1}'$) of $V$ (resp. $X$) around the axis of $B_1$ (resp. $B_{k+1}$). Once these target rotation angles are determined, the points of the initial segment $UV$ (resp. $YY$) are rotated in turn around the axis of the cone by interpolating the rotation angle between $0$ and $\theta_k'$ (resp. between $0$ and $\theta_{k+1}'$) using a function of the normalized distance to $U$ (resp. $Y$) on the initial segment $UV$ (resp. $YY$). As for the twist rotation, we propose to use a cubic interpolation of the angle, but once again it is possible to use another profile, possibly suggested by an artist, learned from experimental data, or resulting from physics-based criteria. Thus, the point $q(d)$ with normalized distance $d$ to point $U$ on segment $UV$ is rotated by an angle $\theta_k(d) = -2\theta_k' d^3 + 3\theta_k' d^2$ around the axis of $B_1$. The same is done for the points of the initial segment $YY$, by using their normalized distance $d$ to $Y$ and by using it.

Figure 7: The right bone is rotated around its axis by the angle $\tau_{max}$. The baseline before (green) and after (red) the twist are superimposed. The right image shows the cubic profile of the twisting angle applied to the points: $\tau(d) = -2\tau_{max}d^3 + 3\tau_{max}d^2, d \in [0,1]$.

Figure 8: Positions of baseline segments before and after a joint is bent. Circles of tangency on $B_k$ and $B_{k+1}$ are represented in blue and red respectively.

Figure 9: Baseline deformation after bend rotation of joints. The rotated segments (in blue) are superimposed with the deformed baseline (in yellow): (a) bend of a single joint between two bones, $U = U'$ and $Y = Y'$ (b) bend of two consecutive joints.
interpolate the rotation angle $\theta_{k+1}(d)$ between 0 and $\theta^V_k$. The resulting deformation of the segment is illustrated in Figure 7.

The curves $UV'$ and $XY'$ obtained by deforming the segments $UV$ and $XY$ are still included on the cone part of their respective bones. If they do not intersect, they can be connected by a circular arc since $V'$ and $X'$ belong to a common plane of the sheaf of planes passing through $A_k, A_{k+1}$. If they intersect, we crop the parts of the curves that get inside the other cone, and we connect the truncated parts.

**Deformation of a complete baseline on a chain of bones.** The deformation of the segments composing a baseline is done by sequentially processing the bones composing a chain, as is always the case when positioning the skin of an articulated body.

The bend of a joint distorts the segments on both its incident bones. If rotations occur at both joints bordering a bone, there is a combined effect on the common bone with a target position on each side for the endpoints of segments being deformed. However, what is said above can be adapted: we move the points of the segments by interpolating a rotation angle around the axis of the cone between two target angles (see Figure 9).

If a bone is also affected by a twist, there is a combined effect of the bend and the twist on the segments of the bone. The twist is simply handled by an additional rotation of one of the endpoint, corresponding to adding the angles $\tau_{\text{maj}}$ to the actual target angle. Then, we proceed by cubic interpolation of the angle between the two target angles for rotating the segment inner points around the axis of the bone.

The deformed segments are completed with circular arcs in the spherical convex parts of the joints, and they are truncated where they intersect in the concave parts of the sphere.

If the transformations of the bones of the skeleton are pre-computed first, the relative displacement of the points above the bones can benefit from parallelism.

**Deformation of baselines at the junction between several chain of bones.** If a bone is bent at a common junction with other chains of bones, we propose to locally deform the baselines by establishing a correspondence between the baselines before and after skeletal deformation. This can be done by giving a normalized coordinate $c$ to each baseline of a given type, based on the proportion of angle crossed by the baseline on the pivot circular arc associated with the pair of its corresponding bones.

Evolution of the detail direction field over the deformed baselines. After the movement of the skeleton, the detail directions above a deformed baseline can be constructed without knowledge of the original baseline. It depends only on the position of the points on the surface of the sphere-mesh. This amounts to compute portions of new baselines as they would be constructed with the new position of the skeleton and to construct the direction field above the segments and circular arcs of this new bundle of baselines. This is illustrated in 2D in Figure 13 and in Figure 14.

3.6. *Base-points and points after transformation*

**Base-points displacement.** When the skeleton moves, the baseline of a point $p$ moves along with the bones. The base-point $b_p$ moves along with this baseline, but it also displaces along the deformed baseline to deal with the enlargement or shrinkage of the baseline parts between two consecutive joints. More precisely, a baseline is divided into sections by anchor points ($S_1$ and $S_{k+1}$ in Figure 13) that were placed at the intersection with the separator planes (on the circular arcs of the baseline or at the intersection between two baseline segments). After movement
original shape and the relative abscissa of its base-point $b_p$ with respect to bordering anchor points on the baseline. This portion of the baseline is further deformed with the skeleton, and the new position of $b_p'$ is computed before reporting the detail in the direction corresponding to $b_p'$.

**Point set skinning algorithm.** The full process of point set skinning is summarized in Algorithm 1.

| Algorithm 1 Pose and anatomy change using baseline skinning |
|---|
| **Input:** Sphere-mesh model registered to the input point set $P$. Target pose and anatomy. **Output:** Point set $P'$ corresponding to the target changes. 1: Define for each point $p \in P$ its base point $b_p$, baseline $b$ on the registered sphere-mesh and compute $h$ and $t$ (ratio of curvilinear distance to bordering anchor points) 2: Deform the baseline $b$ into $b'$ using the target pose and anatomy 3: Determine $b_p'$ and its corresponding $n_b'$ by reporting $t$ on baseline $b'$ 4: Lift point $b_p'$ to the skin surface using $h$ and $n_b'$ |

**3.6.1. Modulation of detail amplitude $h$**

If the shape we consider is an offset surface above the sphere-mesh ($h$ constant everywhere), we would like it to remain an offset surface after deformation. This amounts to a local principle of volume preservation which was not addressed in our approach (see green details in Figure 16b). To introduce this principle it is sufficient to modulate the amplitude of the detail $h$ with a function of the angle between the direction of detail $n_b$ and the normal to the sphere-mesh (see red details in Figure 16b). The normalized value $h'$ is computed by:

$$ h' = \frac{\sin \theta}{\sin \theta_b} h $$

where $\sin \beta$ (or $\sin \beta'$) represents the angles of the detail direction $n_b$ (or $n_b'$). It is this modulated detail that is reported over the base-point after movement of the skeleton. $\sin \beta = 1$ when a base-point is on the sphere part or on the convex part of the sphere-mesh model.
3.6.2. Missing data when unfolding a joint

When a joint is initially folded like in Figure 17a and we want to unfold it, we sometimes lack information because it was hidden in the fold (green area in Figure 17a). Figure 17b shows the position of points on the baseline if we unfold the joint by taking into account the hidden area: a new zone (segment $S_1S_{k+1}'$ and segment $S_{k+1}'S_2'$) appears on the baseline after unfolding, for which we lack detail information. This phenomenon appears only in the concave part of a joint being unfolded. This can be solved by an inpainting process, which is a difficult task for surfaces. When taking into account only the visible part of the baseline (as it was done in previous subsections), we adopt a simpler strategy by not taking the hidden fold into account: we simply extend the two visible parts of the baseline over the area to be inpainted. The problem is that the junction may be unpleasant after reporting the detail since we stitch together details that may correspond to remote areas on the surface of the object (Figure 17c in purple zone). Therefore, we propose a gradual Gaussian smoothing of this 1D area to reduce the small visual artifact which occurs when the starting position was too bent. We apply this smoothing process only in the context where we have a severe fold that possibly hides information. For example, in Figure 18 we gradually apply this smoothing process only for base-points on segment $S_1 S'$ and segment $S' S_2$. $S_1$ and $S_2$ represent the positions of the initial point $S$ on the left bone and the right bone before unfolding the joint. The resulting height amplitude of $p$ is thus:

$$h(b_p') = \frac{1}{cst} \sum_{b_p' \in S} \exp(-\frac{||b_p' - b_p||^2}{2\sigma(b_p')^2})h(b_p')$$

with $cst$ a weight normalizing factor. $H$ is a subset of $b_p'$’s neighbors’ base points shown in green zone in Figure 18. The value of $\alpha(b_p')$ depends on the position of point $b_p'$; it becomes smaller when $b_p'$ is close to the boundary ($S_1$ or $S_2$ in our example) of the smooth zone: $\alpha(b_p') = \frac{b_p'^3}{3}$ if $b_p'$ is on the segment $S_1 S_{k+1}'$ or $\alpha(b_p') = \frac{b_p'^3}{3}$ if $b_p'$ is on the segment $S_{k+1}' S_2$.

Let note that in contrast, when we bend a joint, we choose the approach which consists in leaving the detail visible and pushing it in its entirety outside the fold. But we could very well set thresholds beyond which we would choose to hide part of the detail inside the fold.

4. Results

In this section, we show the performance of our baseline skinning method on synthetic and real point sets. We implemented our algorithm in C++, using OpenMP for computing baselines in parallel. All experiments are run on an Intel Core i7-4790K CPU @ 4.00GHz.

We first test our algorithm on synthetic data. We generated a set of points lying on the same baseline and used a chain of three bones. The attachment of points to bones is shown in different colors: red, green and blue. In Figure 19, we show original points, their base points and detail directions. For different bone motion, we show the initial position (first column), our baseline skinning results by linear interpolation (2nd column) and by cubic interpolation (3rd column) for both $\alpha_p$ and $\alpha_d$. We show the results for a folding motion (first and second row), and an unfolding motion (third and fourth rows), for baselines situated in convex or concave parts after deformation. In all these synthetic problems our baseline skinning method performs well. The skinning result can also be further adjusted by changing interpolation schemes. Since our detail direction is not orthogonal to the bones in concave parts, we avoid the detail direction intersecting near the joints. Then we compare our method with LBS, dual quaternion skinning and skinning method presented in [12] on a stripy synthetic surface ???. We show the results of twisting, bending and re-bending to initial position. Our method gives a good performance for all these movements.

We selected a set of statues from various sources:

1. Dragon, Scan data of a dragon sculpture
2. Dancer with Crotales, Louvre Museum
3. Aphrodite, Thorvaldsens Museum
4. Dancing Faun, Pompei excavations
5. Saint John the Baptist, Ny Carlsberg Glyptotek

While the 'Dancer with crotales' is a raw point set [38]. The other models are point sets sampled on meshes extracted from the Sketchfab website. We show original point set of statues in Figure 21.

| model               | number of points | Ours Fu et al. | DQ | LBS |
|---------------------|------------------|----------------|----|-----|
| Dragon              | 532 067          | 0.48s 0.62s    | 0.34s 0.49s |
| Dancer with Crotales| 539 136          | 1.32s 1.4s     | 0.21s 0.27s |
| Aphrodite           | 461 688          | 0.31s 0.82s    | 0.22s 0.31s |
| Dancing Faun        | 518 037          | 0.63s 1.38s    | 0.22s 0.28s |
| Saint John the Baptist | 519 483        | 0.48s 1.68s    | 0.25s 0.31s |

Table 1: Execution times of different methods for the point sets.

Figures 22, 23, 24, 25 and 26 show our baseline skinning results compared with common skinning algorithms. Compared with Linear blend skinning or Dual-quaternion skinning, our baseline skinning method improves the collapse or bulge effect.
at joint and gives a more reasonable result for the point set. This improvement can be observed especially at joints which move a lot, such as the bending left arm of Dancer with Crotalas (Figure 22), the left leg and the left arm of Aphrodite (Figure 23), the arms of the Dancing Faun (Figure 24) and the right arm and the right leg of the Saint John the Baptist (Figure 25). Compared with the method in [12], which can also improve these artefacts, our method preserve the information of the original point set better. For example, we can better observe the sculpture texture on the right arm of Saint John (Fig. 25) with our skinning result than the result by the method of [12]. Furthermore, the method in [12] does not show points that are hidden inside the sphere-mesh model after deformation and creates a discontinuity for the point set even if the skinning result is visually pleasant. Figure 26 shows a multi-layer point set example. Our baseline skinning method gives the best result among four methods, especially at the head and the tail where the details are complicate. We do not set bones for the claws so we consider the claws as a multi-layer points over the dragon’s body. We can observe the cracked horns in the first point of view and a cracked back claw in the second point of view for the LBS and dual quaternion skinning methods. The cracked effect can be resolved by adjusting the influence weight, but the result is not pertinent (see the third one in Figure 26c). It is difficult for LBS and dual quaternion method to deal with the multi-layer data. The method of [12] performs better for multi-layer data than LBS and dual quaternion skinning method. But it also has a cracked claw because they use orthogonal projection when they compute their height field. The orthogonal projection induces a discontinuity at concave parts of the bones. As far as computation times are concerned, Table 1 shows times of the four methods. Our method is faster than the method of Fu et al. [12] and is similar to Dual-quaternion skinning and Linear Blend Skinning.

We compare our method with the skinning method presented in the paper of original sphere-mesh model [14]. Although they use the sphere-mesh model as the skeleton of the skinning, the artefacts of the classical skinning methods remains. Our method does not lose the volume and preserves well the original connectivity (See the stripes in Figure 27d.).

Limitations. Our baseline skinning method has several limitations. First, an unfolding motion raises a missing data problem since we do not have a mesh of the complete body, such as for the right arm of the Dancer (fourth column in Figure 22a) or the left arm of Saint John (second column in Figure 23a), to alleviate this limitation, one could use a patch-based 3D inpainting process or a 1D inpainting process restricted to baselines. Another limitation is that our skinning method can only deal with points influenced by a single chain of bones and cannot be easily adapted to deal with points influenced by several different bone chains. This limitation can have an impact on examples such as the Dancer’s dress which should be influenced by the motions of both legs. However, this case is clearly one where the very hypothesis of a sphere-mesh model based skinning reaches its limits, it would require cloth modeling and animation techniques.

5. Conclusion and perspectives

We introduced a novel baseline skinning method for point sets of articulated bodies. Our method does not require a weight computation for each point and gives a realistic skinning result when changing the pose and morphology of the articulated body, even when there is no information on the underlying muscles. As a future work, we plan to design displacement fields above baselines to mimic muscles or to take into account additional folds of the skin. We also want to investigate how we can associate different bone chains in order to deal with points which are influenced by three or more bones. It would also be necessary to improve the treatment of the deformations of the junctions because for the moment the proposed solution will involve tightening/stretching effect between neighbouring baselines. We provide a solution to the case where an inpainting problem is raised in concave areas at unfolding/unbending of a joint. However, when the problem is too severe, it can be interesting to develop an other approach, such as patch-based approaches.

Acknowledgments

This work was funded by the e-Roma project from the French Agence Nationale de la Recherche (ANR-16-CE38-0009). The Dancer with crotalas model is a point set of the Farman Dataset [33]. The other statues data are sampled on meshes from the Sketchfab website: the Dragon model is courtesy of Sketchfab user ”3D graphics 101 “, the Dancing Faun model is courtesy
Figure 20: In each sub-figure from up to down: initial position, twisting result, bending result, re-bending to initial position result. We show the weight distribution for the right bone in the first line in sub-figure (b), (c) and (d). We used the same Gaussian weight computation introduced in [12] for (b), (c) and (d). The support of the Gaussian weight is anisotropically reduced in (d).

Figure 21: The original point sets of statues.

References

[1] N. Magnenat-thalmann, R. Laperrine, D. Thalmann, U. D. Montréal, Joint-dependent local deformations for hand animation and object grasping, in: In Proceedings on Graphics interface ’88, 1988, pp. 26–33.
[2] L. Kavan, S. Collins, J. Zára, C. O’Sullivan, Geometric skinning with approximate dual quaternion blending, ACM Trans. Graph. 27 (2008) 105:1–105:23.
[3] A. Mohr, L. Tokheim, M. Gleicher, Direct manipulation of interactive character skins, in: I3D, ACM, 2003, pp. 27–30.
[4] J. P. Lewis, M. Cordner, N. Fong, Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation, in: ACM SIGGRAPH, 2000, pp. 165–172.
[5] M. Alexa, Linear combination of transformations, Proceedings of the 29th annual conference on Computer graphics and interactive techniques.
[6] N. Thalmann, F. Cordier, H. Seo, G. Papagiannakis, Modeling of bodies and clothes for virtual environments., Vol. 0, 2004, pp. 201–208. doi:10.1109/CW.2004.47
[7] B. Le, J. Hodgins, Real-time skeletal skinning with optimized centers of rotation, ACM Transactions on Graphics 35 (2016) 1–10. doi:10.1145/2897624.2925959
[8] X. C. Wang, C. Phillips, Multi-weight enveloping: Least-squares approximation techniques for skin animation, in: SCA, 2002, pp. 129–138.
[9] L. Kavan, J. Zára, Spherical blend skinning: A real-time deformation of articulated models, in: I3D, ACM, 2005, pp. 9–16.
[10] A. Jacobson, O. Sorkine, Stretchable and twistable bones for skeletal shape deformation, ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA) 30 (6) (2011) 165:1–165:8.
[11] L. Kavan, O. Sorkine-Hornung, Elasticity-inspired deforming character articulation, ACM TOG 31 (2012) 196:1–196:8.
[12] T. Fu, R. Chaine, J. Digne, Anatomy Changes and Virtual Restoration of Statues, in: Eurographics Workshop on Graphics and Cultural Heritage, The Eurographics Association, 2020.
[13] I. Baran, J. Popović, Automatic rigging and animation of 3d characters, ACM TOG 26 (3).
[14] J.-M. Thiery, E. Guy, T. Boubekur, Sphere-meshes: Shape approximation using spherical quadric error metrics, ACM TOG 32 (6).
[15] B. Angles, D. Rehman, M. Macklin, B. Wyvill, L. Barthe, J. Lewis, J. Von Der Pahlen, S. Izadi, J. Valentín, S. Bouaziz, et al., Viper: Volume invariant position-based elastic rods, Proceedings of the ACM on Computer Graphics and Interactive Techniques 2 (2019) 1–26.
[16] O. Dionne, M. Lasa, Geodesic voxel binding for production character meshes, in: SCA ’13, 2013.
[17] A. Jacobson, I. Baran, J. Popovic, O. Sorkine-Hornung, Bounded biharmonic weights for real-time deformation, Communications of the ACM 57 (2014) 99 – 106.
[18] B. H. Le, J. P. Lewis, Direct delta mush skinning and variants, ACM TOG 38 (4) (2019) 113:1–113:13.
[19] T. Ju, S. Schaefer, J. Warren, Mean value coordinates for closed triangular meshes, ACM TOG 24 (2005) 561–566.
[20] P. Joshi, M. Meyer, T. Derose, B. Green, T. Sanocki, Harmonic coordinates for character articulation, ACM TOG 26 (2007) 71.
[21] D. Ali-Hamadi, T. Liu, B. Gilles, L. Kavan, F. Faure, O. Palombi, M.-P. Cani, Anatomy transfer, ACM TOG 32 (6) (2013) 188:1–188:8.
[22] B. Gilles, L. Reveret, D. Pai, Creating and animating subject-specific anatomical models, Computer Graphics Forum 29 (8) (2010) 2340–2351.
[23] S. Saito, Z.-Y. Zhou, L. Kavan, Computational bodybuilding:

of Moshe Caine and the 3 other statues models (Aphrodite, Saint John the Baptist and Old Fisherman) are courtesy of Geoffrey Marchal.
Figure 22: Comparison of different skinning methods on the Dancer with Crotales point set. We show close-ups from the second to the fourth row. (a) Pose change with our baseline skinning method, (b) Pose change with Linear blend skinning, (c) Pose change with dual quaternion skinning, (d) Pose change with the method of [12].

Anatomically-based modeling of human bodies, ACM Trans. Graph. 34 (4).

[24] A. E. Ichim, P. Kadlec, L. Kavan, M. Pauly, Phace: physics-based face modeling and animation, ACM Trans. Graph. 36 (2017) 153:1–153:14.

[25] P. Kadlec, A.-E. Ichim, T. Liu, J. Krivanek, L. Kavan, Reconstructing personalized anatomical models for physics-based body animation, ACM Trans. Graph. 35 (6).

[26] R. Vaillant, L. Barthe, G. Guennebaud, M.-P. Cani, D. Rohmer, B. Wyvill, O. Gourmel, M. Paulin, Implicit skinning: real-time skin deformation with contact modeling, ACM TOG 32 (4) (2013) 125:1–125:12.

[27] W. V. Funck, H. Theisel, H. Seidel, Volume-preserving mesh skinning, in: VMV, 2008.

[28] D. Rohmer, S. Hahmann, M. Cani, Exact volume preserving skinning with shape control, in: SCA ’09, 2009.

[29] A. Angelidis, K. Singh, Kinodynamic skinning using volume-preserving deformations, in: SCA ‘07, 2007.

[30] V. Roussellet, N. A. Rumman, F. Canezini, N. Mellado, L. Kavan, L. Barthe, Dynamic implicit muscles for character skinning, Comput. Graph. 77 (2018) 227–239.

[31] P. Sloan, C. F. Rose, M. Cohen, Shape by example, in: I3D ’01, 2001.

[32] R. Y. Wang, K. Pulli, J. Popovic, Real-time enveloping with rotational regression, in: SIGGRAPH 2007, 2007.

[33] A. Mohri, M. Gleicher, Building efficient, accurate character skins from examples, ACM SIGGRAPH 2003 Papers.

[34] P. Koy, D. L. James, D. Pai, Eigenskin: real time large deformation character skinning in hardware, in: SCA ’02, 2002.

[35] F. Policarpo, M. Neto, J. Comba, Real-time relief mapping on arbitrary polygonal surfaces, in: SIGGRAPH 2005.

[36] J.-M. Thiery, E. Guy, T. Boubekeur, E. Eisemann, Animated mesh approximation with sphere-meshes, ACM Trans. Graph. 35 (2016) 30:1–30:13.

[37] T. Fu, R. Chaine, J. Digne, Fakir: An algorithm for revealing the anatomy and pose of statues from raw point sets, Computer Graphics Forum 39.

[38] J. Digne, N. Audfray, C. Lartigue, C. Mehdi-Souzani, J.-M. Morel, Farman institute 3d point sets-high precision 3d data sets, IPOL 1 (2011) 281–291.
Figure 25: Comparison of different skinning methods on the Saint John the Baptist point set. We show close-ups from the second to the fourth row. (a) Pose change with our baseline skinning method, (b) Pose change with Linear blend skinning, (c) Pose change with dual quaternion skinning, (d) Pose change with the method of [12].

Figure 26: Comparison of different skinning methods on the dragon point set under two points of view. (a) Skinning weights computation [14], (b) Sphere-mesh skinning [14] by LBS, (c) Sphere-mesh skinning [14] by DQS, (d) Baseline skinning result.

Figure 27: Comparison with the weight-based skinning approach using sphere-mesh skeletons [14]. The movement of the right bone is a combination of bend and twist.

Figure 28: Comparison of different skinning methods on the dragon point set under two points of view. (a) Baseline skinning method, (b) Linear blend skinning, (c) Dual quaternion skinning, (d) Skinning method of [12].