

Learning Skeletons for Shape and Pose

Nils Hasler*
MPI Informatik

Thorsten Thormählen
MPI Informatik

Bodo Rosenhahn
University of Hannover

Hans-Peter Seidel
MPI Informatik

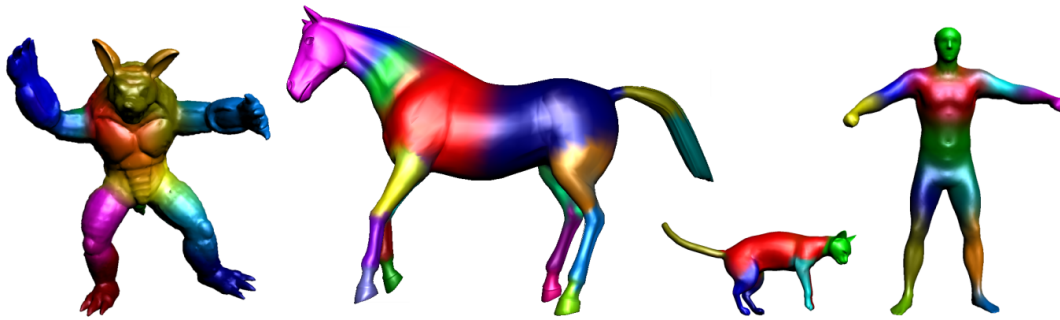


Figure 1: Skeletons can be extracted automatically from examples for a variety of different models (like bipeds or quadrupeds shown here). Each colour shows the area of influence for a particular bone of the skeleton.

Abstract

In this paper a method for estimating a rigid skeleton, including skinning weights, skeleton connectivity, and joint positions, given a sparse set of example poses is presented. In contrast to other methods, we are able to simultaneously take examples of different subjects into account, which improves the robustness of the estimation. It is additionally possible to generate a skeleton that primarily describes variations in body shape instead of pose. The shape skeleton can then be combined with a regular pose varying skeleton. That way pose and body shape can be controlled simultaneously but separately. As this skeleton is technically still just a skinned rigid skeleton, compatibility with major modelling packages and game engines is retained. We further present an approach for synthesizing a suitable bind shape that additionally improves the accuracy of the generated model.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Hierarchy and geometric transformations

Keywords: skeleton estimation, linear blend skinning

1 Introduction

Creating animatable models is an important step both in the development of interactive games as well as in high-end movie productions. Although a number of deformation techniques have been developed in the last years, the most common method especially for interactive graphics is skeleton based deformation. Here, every vertex of a surface model is attached to the bones of a hierarchical skeleton, which frequently mimics the physiological skeleton of the modelled entity. By specifying transformations for every bone of

the skeleton an artist can deform the model and create animations. The vertices on the surface move along with the bone they are attached to. By allowing a vertex to be attached to more than one bone and interpolating between the transformed vertex positions, smooth transitions on the surface between bones can be achieved. Despite its drawbacks pointed out e.g. by [Weber 2000] and an increasing number of skeleton-less alternatives, linear blend skinning (LBS) also known as skeletal subspace deformation (SSD), single-weight enveloping or just skinning still remains the most common interpolation technique. It is available in all major 3D modelling packages, used in many games, and applied in movie productions. Unfortunately, the generation of skeleton hierarchy and skinning weights (interpolation weights) is a labour intensive task, normally performed by specialised artists (character riggers).

Consequently, several methods have been proposed that automatically embed a skeleton into a given mesh. However, since the information that can be gathered given only one example is limited, either a skeleton has to be provided by the user [Baran and Popović 2007], which is only adapted to the specific mesh, or a heuristic approach is used to segment the mesh into parts, which are assumed to correspond to bones of the skeleton [Tschirren et al. 2002]. Furthermore, it is very hard to compute smooth skinning weights given only one example.

Here, we present a method that automates the process of skeleton generation and skinning. Given a small number of examples in different poses, the proposed method generates a bone hierarchy, joint positions, and skinning weights as used in linear blend skinning. The examples can either be acquired from real objects by using a 3D scanner or can be created manually with a 3D modelling package.

Although [He et al. 2009] describe a method that allows transferring a skeleton from one entity to another, e.g. from a horse to a cat, it is not possible to take examples from different entities into account during the construction of the skeleton. In contrast, with the method we propose several entities, e.g. different scanned human subjects, can be taken into account during optimisation because we are only considering relative deformations. So, if one reference per entity/subject is used, only pose dependent deformations are visible to the optimisation procedure. On the other hand, with the proposed method, it is also possible to compute a skeleton, given a set of different subjects in approximately the same pose. Using

*e-mail: hasler@mpi-inf.mpg.de

Copyright © 2010 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.

I3D 2010, Washington, DC, February 19 – 21, 2010.
© 2010 ACM 978-1-60558-938-1/10/0002 \$10.00

this approach a classic skeleton can be automatically extracted that describes shape changes rather than pose deformations. By merging these two skeletons a combined skeleton can be generated that describes shape as well as pose deformations.

Another problem, that has, to the best of our knowledge, not been addressed in the literature, is the generation of a suitable bind shape (dress pose), i.e. creation of the shape that is used as the basis for all deformations. In the ideal case, it does not matter which pose the bind shape is in. In reality, however, it is important to choose a pose that is easy to change for the modeller. Ideally the pose is symmetric. It also makes sense to choose a pose that it is as close as possible to the poses the model is expected to perform because artefacts tend to be the more prominent the farther a deformation deviates from the bind shape. We describe a method for first computing an initial bind shape and a means to further refine it, given the final skeleton, skinning weights, and the input examples.

Contributions

- We estimate a skeleton that either describes shape variations or variations in pose. The combination of the two, allows us to control body shape and pose independently of each other, while retaining compatibility with current modelling packages and game engines.
- In contrast to other methods we are able to combine the information from example sets from different entities/subjects to improve the stability of the method.
- Additionally, an approach for synthesising an optimised bind shape is presented.

The paper is structured as follows. After presenting related work (Sec. 2), the proposed method is introduced (Sec. 3), results are presented (Sec. 4), and a summary is given in Section 5.

2 Related Work

Since the skeleton fitting and skinning process is labour intensive and time consuming, a number of algorithms have been published that address the problem. Since the issue is multilayered some publications address only part of the topic. In the following, we first describe skeleton extraction using a single input image, skeleton extraction from several examples, and then briefly introduce other methods for describing pose and shape simultaneously.

Several approaches have been presented that estimate a skeleton, given only a single input example. Since the information inherent in a single input mesh is limited, the approaches are either not interested in obtaining a skeleton suitable for animation or rely on additional information such as an example skeleton that is adjusted to fit to the data. In the former case the primary objective can be shape matching [Sundar et al. 2003], segmentation [Au et al. 2008; Katz and Tal 2003; Tschirren et al. 2002], or topology analysis [Bai and Liu 2007], to name just a few.

Allowing animation of a single input mesh without skilled user input is the clear aim of [Baran and Popović 2007]. They provide a skeleton in a similar pose along with the input mesh to the algorithm. This skeleton is then moved and morphed such that it fits as nicely as possible into the provided mesh. By discretising the medial surface using packed spheres the problem of locating the joint centres is turned into a graph matching problem, which can be solved efficiently. Skinning weights are computed by solving a differential equation, which is modelled in analogy to heat diffusion. The results are convincing but the algorithm is unable to work with non-standard articulation unless a special skeleton is provided.

In a recent contribution, [Xu et al. 2009] analyse 3D models which consist of several parts. They are able to identify joints between rigidly moving parts, and the degrees of freedom between the different components. Combining an inverse kinematics solver with a grid-based deformation method results in a simple posing and deformation system. Unfortunately, the resulting articulation is not compatible with commonly used animation systems.

An interesting approach for improving given skeleton based animations was presented by [Mohr and Gleicher 2003]. They propose to add additional scaling transformations coupled to the rotation of bones to counter the most common artefacts present in linear blend skinning. Of course, by augmenting LBS their approach loses compatibility.

[James and Twigg 2005] were the first to introduce a fully automatic method for generating both a skeleton, and the corresponding skinning weights, given a sparse set of example meshes. They used mean-shift clustering to detect near-rigid structures, and allow their bones to perform any 3×3 transformation including rotation, scaling, and shearing, which is not compatible with most modelling systems. Skinning weights are estimated using non-negative least squares to solve for the best weights of every vertex.

Similarly, [Schaefer and Yuksel 2007] presented an approach for estimating a LBS compatible skeleton given a set of examples. They also solve the problem of finding attachment regions for the bones using a clustering method that merges clusters until a specified number of clusters is reached. Skinning weights are then computed by iteratively solving a linear equation system and after each iteration enforcing non-negativity and truncating the number of bone attachments for each vertex to four. Additionally, they use flood fill segmentation to restrict the influence of a bone to a local region. Joint positions are computed by exploiting knowledge about the skeleton hierarchy. The skinning weights of two connected bones normally overlap and thus provide information about the probable location of the joint.

More recently, [de Aguiar et al. 2008] recommended to use spectral clustering to segment the mesh into approximately rigidly moving parts. The proposed cost function penalises vertices, whose distance changes within the example set. Additionally, connections between vertices with large overall distances are penalised to force them to be clustered into different bones. For building the hierarchy they consider only clusters with shared boundaries. If several clusters are adjacent the largest boundary is taken to be the correct one. Skinning is performed as proposed by [Baran and Popović 2007], which was developed to work for a single input model and consequently ignores some of the available information.

Very recently, [He et al. 2009] proposed a method that is able to work with example meshes with different triangulations. Their approach is based on harmonic 1-forms, i.e. they compute Reeb graphs of a harmonic function defined on the given examples. This allows them to match examples independent of their resolutions and triangulations. Joint locations are then determined by finding local maxima in the change in average mean curvature of the graph and refined by solving a linear equation system. Finally, skinning is performed using the method proposed by [Xian et al. 2006].

In a different vein, a number of approaches have been presented that model human shape and pose by various means. Possibly the most prominent approach [Anguelov et al. 2005] performs principal component analysis on the vertex positions of a number of examples in the rest pose. The result is then merged with a skeleton based system for pose deformation. The method is successful but fairly complex as two different representations are used for shape and pose, the results of which have to be merged to generate final output meshes. Based on this approach, parametric models, such as

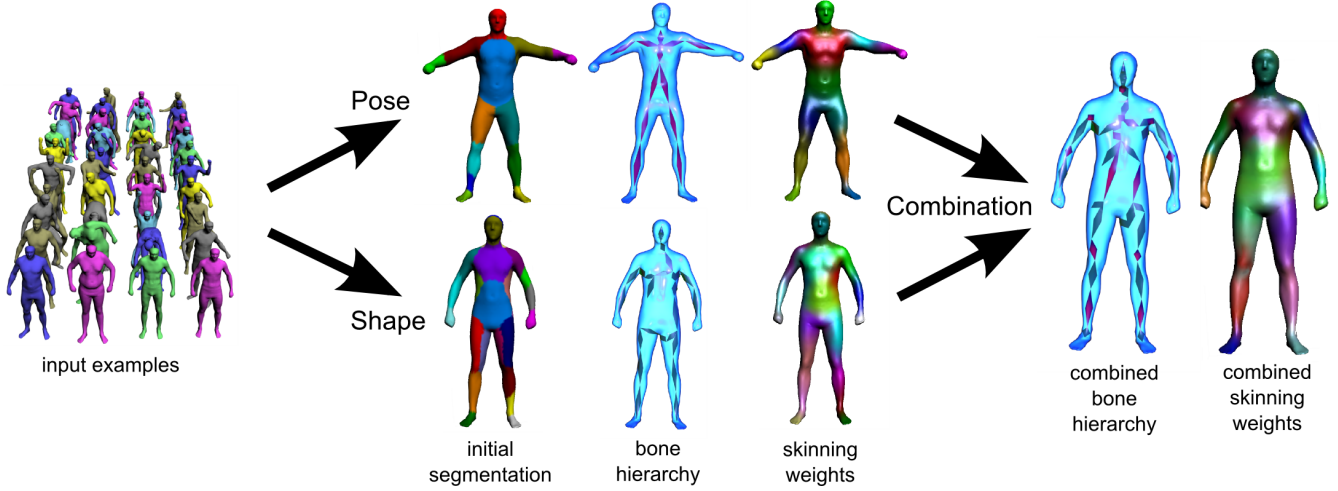


Figure 2: The overall optimisation pipeline is shown. Starting from a set of examples, different subjects in different poses, first an initial segmentation is computed with spectral clustering. Then, by iteratively solving for skinning weights and rotations/translations final skinning weights are generated. The transformations are used to compute the rigidity of potential connections between bones. The minimum spanning tree on this rigidity matrix, is equivalent to the corresponding hierarchy. If both, a pose and a shape skeleton are available, a combined skeleton can be used to control pose and shape of the mesh independently.

[Seo and Magnenat-Thalmann 2003], are able to synthesise body shapes by specifying a small set of semantically meaningful body parameters. The model is computed by training morphing functions for the desired parameters given a set of examples.

A survey of methods for generating animatable human shapes by various means is given by [Magnenat-Thalmann et al. 2003]. Recently, a fully differential model describing pose and shape has been presented by [Hasler et al. 2009]. The representation is uniform in pose and shape but involves solving two equation systems to reconstruct one mesh. In summary, none of the systems relies solely on rigid skeletons to perform shape and pose deformations and are not directly compatible with current 3D modelling tools.

3 Approach

In this section the approach for estimating rigid skeletons given a set of example meshes with identical mesh connectivity is detailed.

In the simplest case the optimisation procedure can be split into five parts. Figure 2 visualises the involved steps. Firstly, a rough segmentation of the mesh into parts belonging to different bones is computed using spectral clustering (Sec. 3.1). Secondly, factorisation leads to initial skinning weights and estimates of the involved bone rotations and offsets (Sec. 3.2). In the third step, the bone hierarchy is computed by constructing the minimum spanning tree using joint location stability as the penalty function (Sec. 3.3). The joint locations are initialised to lie on the plane separating involved bones. An optimisation scheme then improves skinning weights and joint locations (Sec. 3.4). Finally, an optimal bind shape can be synthesized by solving a linear equation system (Sec. 3.5).

If scans from several subjects are given and both pose and shape are to be described by the skeleton, then, in order to preserve orthogonality of pose and shape, initially two skeletons are generated, one for shape and one for pose. These skeletons can be merged in a final step (Sec. 3.6). This allows pose and shape of a mesh to be controlled independently but with a joint representation.

3.1 Initialisation

One of our stated goals is to improve the skeleton estimation by incorporating several different subjects into the computation. The main idea to achieve this goal is that deformations between models are considered rather than using a global criterion. So by grouping the models according to subject, we can limit the observed deformations to pose dependent deformations while gathering information from several subjects.

Initially, the bind shapes for all subjects are computed using the relative rotation encoding proposed by [Hasler et al. 2009]. The encoding represents the orientation of triangles relative to their neighbours and in-plane deformations relative to a canonical template shape. That way, interpolation of arbitrary shapes produces reasonable results, provided the mesh connectivity is the same. Thus, all example models are encoded, the mean is computed, and the resulting mean shape is decoded. This average model preserves both angles between neighbouring triangles and their in-plane shapes as much as possible and serves as the initial bind shape.

We further compute a rough initial segmentation of the mesh into rigid parts. Starting from a given template, we can compute the transformation $\mathbf{T}_{i,e}$ into example e for every vertex i . Vertices that undergo similar transformations are moving rigidly. Considering only the rotational part $\mathbf{R}_{i,e}$ of $\mathbf{T}_{i,e}$ the angle between the transformations for vertices i and l can be computed by converting $\mathbf{R}_{i,e}\mathbf{R}_{l,e}^\top$ into a log-quaternion and computing its magnitude. This is a measure for the non-rigidity of the vertices. Spectral clustering (we use the self tuning variant [Zelnik-Manor and Perona 2005]) of the resulting rigidity matrix leads to an initial segmentation into a specified number of body parts.

3.2 Factorisation

Linear blend skinning makes the assumption that a transformed vertex \mathbf{p}'_i can be represented by the original vertex \mathbf{p}_i , the rotations \mathbf{R}_j for each joint j , and scalar weights $w_{j,i}$, such that

$$\mathbf{p}'_i = \sum_j w_{j,i} (\mathbf{R}_j \cdot \mathbf{p}_i + \mathbf{v}_j), \quad (1)$$

where \mathbf{v}_j is an additional offset that encodes the position of the joint centre. In our context, a number of these equations can be set up for each group of models belonging to one subject. Generally, it is possible to compare every model from a group with every other model from the group. However, if more than a few example models are considered, it becomes computationally intractable to consider all combinations. Since, the primary objective is to optimise the deformation starting from a single bind shape, it is sufficient to consider the transformations from bind shape to examples but not the transformations between different examples.

Since a first weight matrix \mathbf{W} is available from the initial segmentation, we can rearrange Equation (1) and solve for \mathbf{R}_j and \mathbf{v}_j . Unfortunately, the resulting matrices \mathbf{R}_j are not necessarily orthonormal. This can be fixed by SVD based ortho-normalisation [Higham 1986] but the resulting rotations may deviate significantly from the desired results. This situation can be improved by adding additional equations of the form

$$\mathbf{p}_i = \sum_j w_{j,i} (\mathbf{R}_j^{-1} \cdot \mathbf{p}'_i + \mathbf{v}'_j), \quad (2)$$

where \mathbf{v}'_j are additional offset vectors but \mathbf{R}_j remains the same because $\mathbf{R}_j^\top = \mathbf{R}_j^{-1}$. The resulting matrices are very close to symmetric and the resulting ortho-normalisation is much closer to a reasonable rotation. Enforcing true rotation matrices leads to a non-linear cost function. Thus, the current result for \mathbf{R}_j , \mathbf{v}_j , and \mathbf{v}'_j can be optimised further with the Levenberg-Marquardt algorithm [Marquardt 1963].

In a second step we consider the rotations fixed and optimise the weight matrix. This can be done by solving a linear system e.g. with a non-negative least squares solver. However, this approach does not result in localised regions of influence of the bones. I.e. a bone may influence spatially distant regions, which can lead to unsightly artefacts. It also leads to many nonzero weights. [Schaefer and Yuksel 2007] propose to perform segmentation on the weight map and keep only the most influential patch for each bone. Additionally, they cut off all nonzero influences beyond the strongest four. This approach is easy to implement and seems to work. Yet, it may be desirable to have one joint control two limbs if they are always moving synchronously and the limit of four influences seems, apart from efficiency reasons when implementing LBS on graphics hardware, arbitrary.

In contrast, we opt to introduce an additional constraint into the optimisation. Namely, the $L1$ -norm of the weight matrix, which induces sparseness, is minimised simultaneously. This optimisation target is obviously in conflict with the constraint $\sum_j w_{j,i} = 1$. However, as the sum of weights constraint is minimised in the least squares sense while sparsity is enforced with the $L1$ -norm, after renormalisation, the result is still more sparse than a non-negative least squares solution. We use the implementation provided by [Zhang 2009]. After about 10 iterations of solving for rotations and weights the method converges and we can continue by computing a hierarchy of bones.

3.3 Hierarchy Generation

The bone rotations and translations computed in the previous step cannot be transformed directly into a skeleton hierarchy because the offset vectors \mathbf{v}_j are not necessarily consistent with the intrinsic joint hierarchy. This can be easily seen if we consider the following: The position of point \mathbf{p}_1 after rotating it by \mathbf{R}_0 around point \mathbf{p}_0 is determined by

$$\mathbf{p}'_1 = \mathbf{R}_0 \cdot (\mathbf{p}_1 - \mathbf{p}_0) + \mathbf{p}_0 = \mathbf{R}_0 \mathbf{p}_1 + \underbrace{\mathbf{p}_0 - \mathbf{R}_0 \mathbf{p}_1}_{\mathbf{v}_0}. \quad (3)$$

The optimisation, performed in Section 3.2, employs Equation (1), and allows \mathbf{v}_j to be chosen freely. This additional degree of freedom effectively allows the optimisation to freely move the joint centres for every example. However, due to the underlying assumption that the 3D models consist mostly of rigidly moving body parts, we can nonetheless use the estimates to compute connectivity and rough positions of the true joint centres. The basic insight that, given two connected bones j and k , the connecting joint centre $\mathbf{j}_{j,k}$ is invariant with respect to the child bone's rotation, leads to

$$\mathbf{R}_j \mathbf{j}_{j,k} + \mathbf{v}_j = \mathbf{R}_k \mathbf{j}_{j,k} + \mathbf{v}_k. \quad (4)$$

Thus, given a number of observations \mathbf{R}_n , \mathbf{v}_n , the joint centre can be determined by solving the overdetermined system of equations

$$(\mathbf{R}_{j,n} - \mathbf{R}_{k,n}) \mathbf{j}_{j,k} = \mathbf{v}_{k,n} - \mathbf{v}_{j,n} \quad (5)$$

for $\mathbf{j}_{j,k}$. Determining the residual error leads to a measure of the connectedness of two bones [Schaefer and Yuksel 2007]. Computing the full connectedness graph and finding the minimum spanning tree [Kruskal 1956] leads to an optimal skeleton hierarchy.

Determining the root node is mostly a question of ease of use for a human modeller. For optimisation purposes it is not really relevant. A human, however, expects the root to be lie near the perceived centre of the model. We accordingly choose the *graph centre* [Harary 1994] as the root node.

3.4 Joint Position Estimation

The main challenge when determining joint centres lies with hinge joints because the joint centre of a hinge joint can lie anywhere on the hinge axis. Of course in real world examples with measurement inaccuracies and non-rigid deformation this case cannot be detected by Eigenanalysis of the rotation matrices, which would be a good indicator in the ideal case. The joint centres computed with Equation (5), however, exhibit the described artefact. The points lie anywhere on the approximate hinge axis. Again, the skeletons are fully functional but the human modeller expects the bones to lie within the surface of the model. So we initialise the joint centres by placing them on the interface between the two involved bones using the equation provided by [Schaefer and Yuksel 2007]

$$\mathbf{j} = \frac{\sum_i \min(w_{i,1}, w_{i,2}) \mathbf{p}_i}{\sum_i \min(w_{i,1}, w_{i,2})}, \quad (6)$$

where \mathbf{p}_i are all vertices with nonzeros weights for the two bones. Afterwards, the positions \mathbf{j} are refined with a Levenberg-Marquardt gradient descent scheme, optimising Equation (1), which can be expressed as a non-linear function of \mathbf{j} . It also proved beneficial to repeat the skinning weight optimisation from Section 3.2 after updating the joint positions.

3.5 Bind Shape Synthesis

In a final step, the bind shape of the model can be optimised. So far, all existing approaches have used an arbitrary example as the bind shape. This model is typically chosen manually. In contrast, in our method, the initial bind shape is the average model generated using the relative rotation encoding (cf. Section 3.1). At this stage, all unknowns of Equation 1 assuming fixed bind shape \mathbf{p}_i and examples \mathbf{p}'_i are determined. But the result has been computed given in a least squares sense. Thus, it is possible to improve the solution by optimising the bind shape. Rearranging and solving the linear system for \mathbf{p}_i results in a bind shape that reduces the reprojection error of the training examples (cf. Figure 3).

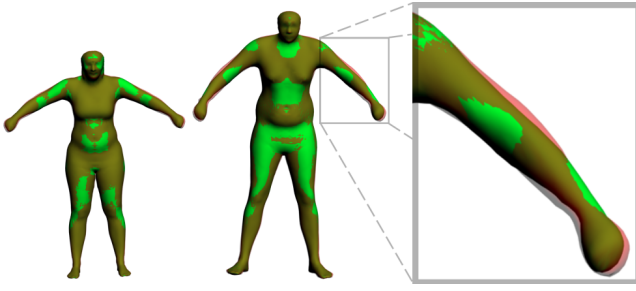


Figure 3: Comparison of the initial bind shape (red) and the improved bind shape (green) for a female and male example (with detail magnification)

3.6 Combining Shape and Pose

The above procedure describes a method for generating a skeleton given a set of example models. With it, we can create skeletons for changing either pose or shape. Yet, it is not feasible to directly create a combined model of pose and shape that separates the contributions of shape and pose. This would be of great importance for easy manual animation of the resulting skeleton. In this section an approach is introduced for generating such a combined skeleton starting from two skeletons, one describing pose and one for shape.

Firstly, a shape deformation skeleton is computed, using models of different persons each scanned in a similar rest pose using the approach described above. The generated skeleton is then used to recreate the subjects in their rest poses. Secondly, these models derived with the shape skeleton serve as bind shapes for the second skeleton, which describes pose variations. Since the two skeletons are coupled via the bind poses, a combined skeleton can be created by performing the two transformations one after the other. So, similar to Equation (1), the combined transformation can be expressed as

$$\mathbf{p}'_i = \left(\sum_j w_{j,i}^s (\mathbf{R}_j^s + \mathbf{v}_j^s) \right) \cdot \left(\sum_j w_{j,i}^p (\mathbf{R}_j^p + \mathbf{v}_j^p) \right) \cdot \mathbf{p}_i. \quad (7)$$

In this formulation every shape bone influences every pose bone. This is undesirable because the resulting graph structure is highly connected, whereas common graphics packages can only handle tree structures. Consequently, it is necessary to simplify the graph to a tree. Of course, this is an approximation to the real structure but since the areas of influence of the bones are highly localised, most bones don't overlap significantly. The corresponding factors can consequently be dropped. Since the weight matrices of the two skeletons describe the areas of influence, it is clear that the multiplication of the two, results in a merit function that clusters bones by their area of influence. By concatenating the two weight functions $\mathbf{W}_c = [\mathbf{W}_s, \mathbf{W}_p]$, a general merit matrix \mathbf{M} can be computed.

$$\mathbf{M} = \mathbf{W}_c^\top \mathbf{W}_c \quad (8)$$

Similar to Section 3.3, the minimum spanning tree leads to a reasonable tree structure fitted into the graph. In the given problem, however, additional constraints have to be respected. Namely, it is essential that the hierarchy of the original skeletons is preserved. A joint a that is a child (direct or indirect) of joint b in the a shape skeleton has to remain a child of joint b in the combined hierarchy. It is admissible, however, to introduce additional nodes between a and b as long as they are from the pose skeleton. The reverse delete algorithm for computing the minimum spanning tree [Kruskal 1956] can be modified to incorporate this type of constraint. Unfortunately, after the modified algorithm terminates,

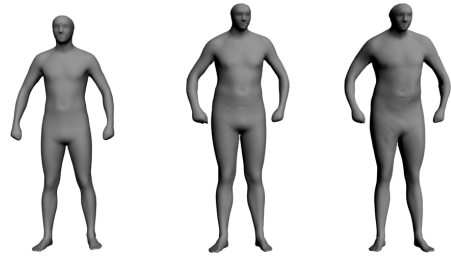


Figure 5: Transforming a bind shape (left) to another shape using a shape skeleton that allows rotation and scale (middle) in comparison with a shape skeleton that allows rotation and translation (right).

some small cycles can remain. These cycles are dissolved by connecting the involved nodes fully and rerunning the above modified reverse delete algorithm. This time, however, only the pose hierarchy is enforced. Since these cycles normally appear in the chest region, near the root of the tree, violations of the shape hierarchy which primarily performs translations and only little rotation are not vital. In the resulting skeletons pose and shape bones frequently alternate because the constraints allow only insertion of nodes from the respective other skeleton (cf. Fig. 4).

3.7 Translation vs. Scaling

Since the articulated motion of humans can be described well by distinct rigid body motions with blending, skeleton based systems describing rotation and translation for every bone are very successful. Shape changes on the other hand, intuitively, can be explained more easily by a hierarchy of scale and rotation transformations. Thus, for shape describing skeletons we incorporate additional terms into Equation (1)

$$\mathbf{p}'_i = \sum_j w_{j,i} (\mathbf{R}_j \mathbf{S}_j \cdot \mathbf{p}_i + \mathbf{v}_j), \quad (9)$$

where \mathbf{S}_j is a diagonal matrix and \mathbf{v}_j does not encode any translation beyond the initial offset of a joint relative to its parent. The optimisation strategy described above has to be adjusted accordingly but the necessary changes are straight-forward. Figure 5 shows the difference in quality that can be achieved when computing a shape skeleton with translation or scaling.

A significant disadvantage of incorporating scale into a skeleton hierarchy is that it invariably introduces shearing in dependent nodes during articulation. This effect is not substantial if all examples are in approximately the same pose but can introduce serious artefacts during pose animation. The artefact can be avoided by performing the scaling in the untransformed coordinate system of the bone, i.e. the transformation \mathbf{T}_a of bone a , which is a child of bone b , can be written as

$$\mathbf{T}_1 = \mathbf{R}_a \mathbf{R}_b \mathbf{S}_a \mathbf{R}_b^\top, \quad (10)$$

where \mathbf{R} and \mathbf{S} denote rotation and scale matrices, respectively. Unfortunately, this non-standard approach is not generally supported by standard graphics packages. Since the difference between the shape described by translation and that described by scaling is not significant, as shown in Figure 5, we instead opt to allow translation instead of scaling during shape skeleton construction.

4 Results

In this section experiments are described that show the effectiveness of the approach presented in the previous section. We start by

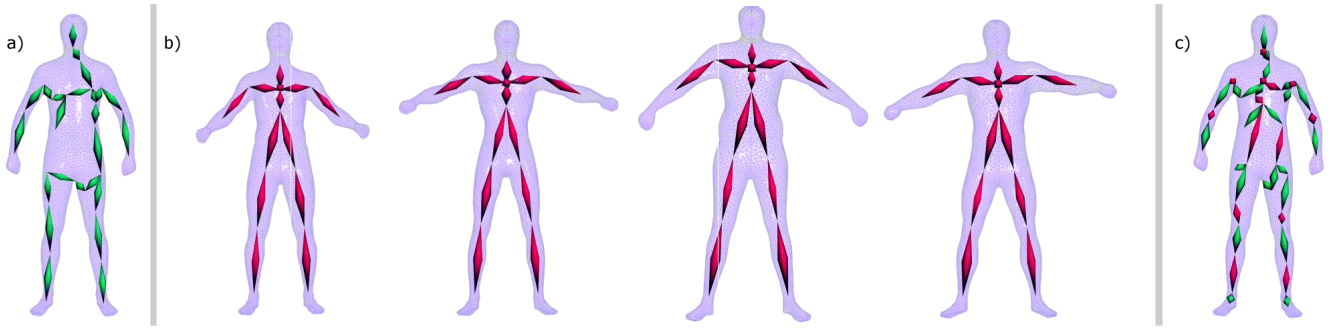


Figure 4: A shape skeleton (a) and pose skeleton for different subjects that were estimated simultaneously (b) are merged into a combined pose and shape skeleton (c). In the combined skeleton the hierarchy of the pose skeleton is enforced. Since shape skeletons are not as sensitive to changes in the hierarchy, the shape hierarchy is allowed to be broken if no overall consistent tree can be found.

showing results exploring the classic pipeline for estimating a pose dependent skeleton (the top branch in Figure 2). Then, new poses of the combined shape and pose model are shown. Animated results are also available in the accompanying video.

As described above, the bind shape is first approximated by the rotation invariant mean of the examples and refined by solving a linear equation system as the last step of the pipeline. A comparison of the two is shown in Figure 3. Although the differences are subtle, slightly more detail is visible in the refined shape and the numerical error is significantly (for the horse on average 11%) smaller.

Final results for the three single entity sets are shown in Figures 7, 8, and 9. They show the refined bind shapes, the input meshes used during optimisation, the estimated skeletons, skinning weights, and a few poses not present in the input sets. Of course, the input sequences used for these three models were generated with similar methods as the technology underlying our approach. It may consequently be unsurprising that it is possible to achieve accurate results with it. The human dataset, however, was obtained by registering 3D scans of real persons to a template mesh. So the assumptions we make (all deformations are based on a rigid skeleton) are only approximately true. The results we obtain are, nonetheless, as good as for the synthetic sequences.

Table 1 summarises the residual root mean squared errors (RMSE) for recreating the input examples as a function of the number of bones for the male and female models, as well as the horse and the cat. As shown, the accuracy is comparable to other recent methods [Schaefer and Yuksel 2007; de Aguiar et al. 2008]. It is also interesting to note that, despite the gross simplification introduced by merging the skeletons (cf. Sec. 3.6), the error for the combined model is not significantly higher than that of the solely shape or pose dependent models.

Since the final model does not use any non-standard techniques, it is possible to load the resulting skinned mesh into common modelling packages. Figure 6 shows the horse model posed in 3D Studio Max.

For symmetric shapes, a user might expect the algorithm to find a symmetric skeleton. Yet, the only symmetrisation we currently perform is, that for the humans, we add mirrored versions of the examples to the input set. The resulting pose skeletons are frequently fairly symmetric but the shape skeletons are not. This is probably related to the fact that, since shape skeletons are mostly controlled by translation, the exact hierarchy of the bones is not as important as for pose skeletons. The undesired translation of a parent node can then be compensated for by translating in the opposite direction, which is not possible for rotations. Nonetheless, it would be possible and easy to enforce symmetric skinning weights, and joint

model	S	N	RMSE (%)				
			10	11	12	13	14
# Bones							
pose (men)	4	80	0.70	0.62	0.61	0.59	0.57
pose (women)	6	120	0.66	0.61	0.60	0.61	0.56
# Bones			10	15	20	25	30
shape (men)	59	118	0.48	0.41	0.37	0.34	0.32
shape (women)	55	110	0.54	0.45	0.41	0.37	0.35
# Bones			34	39	44		
combined (men)	59	190		1.12	0.88	0.96	
combined (women)	55	218			0.88	0.92	
# Bones			10	15	20	25	
horse	1	12	0.90	0.56	0.51	0.38	
# Bones			12	13	14	15	
cat	1	8	1.45	1.20	1.16	1.02	

Table 1: The residual RMSE normalised by the bounding volume diagonal is shown as a function of the number of joints. *S* denotes the number of subjects and *N* the number of examples. The shape model and the combined model allow rotation and translation of the bones only.

positions for either type of skeleton.

Figure 10 shows two shape skeletons, one describing female body shape and one for males and three pose skeletons for different subjects. Each is shown in two significantly different poses/shapes. Morphing shapes and poses is also shown in the video. Exemplary results of the merge of pose and shape skeletons are shown in Figure 11. Here a woman walks and simultaneously changes body shape.

The presented approach is limited in that it can only estimate bones if the relative motion is non-zero in at least one example. If, for example, in a set of meshes the arms of a subject move in parallel, it is impossible for the algorithm to determine that two bones are required. A user may, on the other hand, expect that the prongs of a forklift are controlled by a single bone. Distinguishing these two cases is, in the general case, difficult. An additional limitation of the method is that when combining shape and pose skeletons, the correlations between pose and shape are lost. This effect cannot be avoided unless the commitment to a strict skeleton hierarchy with LBS is dropped.

Despite these limitations, a powerful method for converting a sparse set of examples into a fully rigged kinematic skeleton, has been presented.

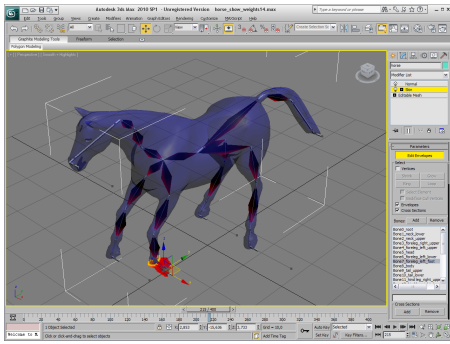


Figure 6: The final model is compatible with current 3D modelling tools and game engines. Here, the horse is posed in 3D Studio Max.

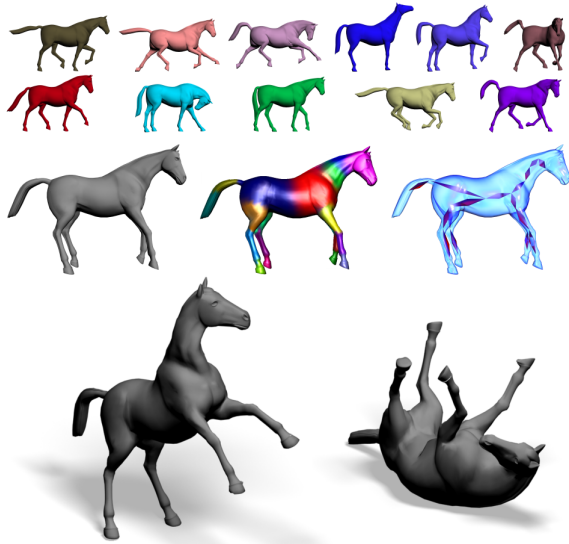


Figure 7: **Top:** input examples, **Middle (left to right):** bind shape, segmentation, extracted bone skeleton, **Bottom:** new poses generated with the extracted skeleton.

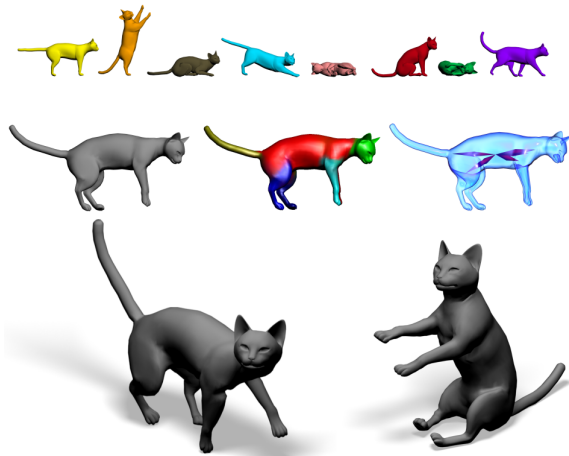


Figure 8: **Top:** input examples, **Middle (left to right):** bind shape, segmentation, extracted bone skeleton, **Bottom:** new poses generated with the extracted skeleton.

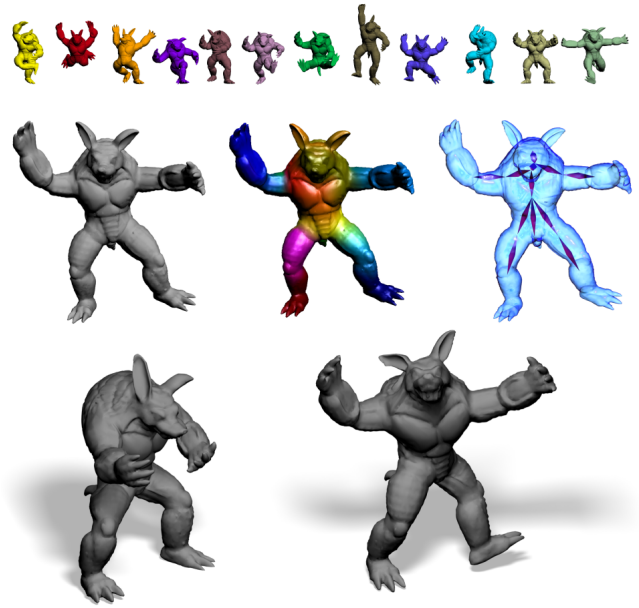


Figure 9: **Top:** input examples, **Middle (left to right):** bind shape, segmentation, extracted bone skeleton, **Bottom:** new poses generated with the extracted skeleton.

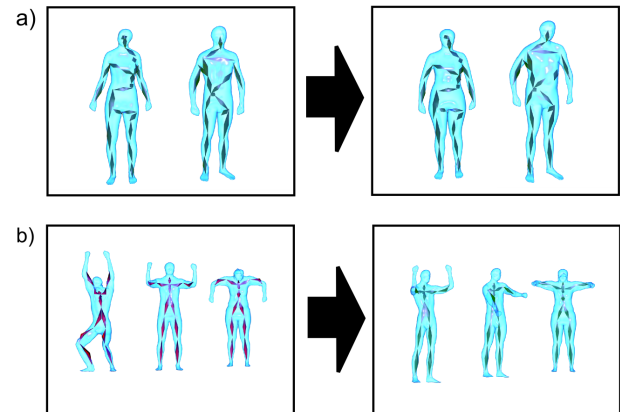


Figure 10: a) changing the shape of the human model with the extracted shape skeleton, b) changing the pose of the human model with the extracted pose skeleton.

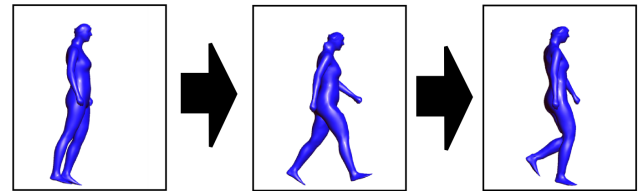


Figure 11: The extracted combined shape and pose skeleton allows independent control of shape and pose. In this example a walking motion is performed while changing the body shape.

5 Summary

We have presented a method for estimating a skeleton, including the hierarchy, joint positions, and skinning weights, given a sparse number of examples, that span the desired space of pose and shape. Unlike other approaches, the presented technique relies solely on rigid skeletons to model pose and shape variations. By combining a pose skeleton and a shape skeleton, it is possible to control the two independently. Another advantage of the presented approach is that it is possible to combine information gathered from several different persons to more robustly train individual skeletons for each of them. Additionally, an approach for computing a bind shape that reduces artefacts such as the candy wrapper effect is presented. The method is evaluated on several datasets including one challenging set of registered 3D scans of over 100 persons in various poses.

Acknowledgements

We would like to thank Robert Sumner for providing the horse and the cat and the aim@shape project for the deformed Armadillo. The work was partially funded by the Cluster of Excellence for Multimodal Computing and Interaction at Saarland University and the Max-Planck-Center for Visual Computing and Communication (BMBF-FKZ01IMC01).

References

- ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: shape completion and animation of people. *ACM ToG* 24, 3, 408–416.
- AU, O. K.-C., TAI, C.-L., CHU, H.-K., COHEN-OR, D., AND LEE, T.-Y. 2008. Skeleton extraction by mesh contraction. In *ACM SIGGRAPH Papers*, ACM Press, New York, NY, USA, 1–10.
- BAI, X., AND LIU, W.-Y. 2007. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 3, 449–462. Member-Latecki, Longin Jan.
- BARAN, I., AND POPOVIĆ, J. 2007. Automatic rigging and animation of 3d characters. *ACM ToG* 26, 3, 72.
- DE AGUIAR, E., THEOBALT, C., THRUN, S., AND SEIDEL, H.-P. 2008. Automatic Conversion of Mesh Animations into Skeleton-based Animations. vol. 27, 389–397.
- HARARY, F. 1994. *Graph Theory*. Addison-Wesley.
- HASLER, N., STOLL, C., SUNKEL, M., ROSENHAHN, B., AND SEIDEL, H.-P. 2009. A statistical model of human pose and body shape. In *Eurographics*, no. 28.
- HE, Y., XIAO, X., AND SEAH, H.-S. 2009. Harmonic 1-form based skeleton extraction from examples. *Graphical Models* 71, 49–62.
- HIGHAM, N. J. 1986. Computing the polar decomposition—with applications. *SIAM Journal of Scientific and Statistical Computing* 7, 4, 1160–1174.
- JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. *ACM SIGGRAPH Papers* 24, 3 (Aug.).
- KATZ, S., AND TAL, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph.* 22, 3, 954–961.
- KRUSKAL, J. B. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematical Society*, American Mathematical Society, vol. 7, 48–50.
- MAGNENAT-THALMANN, N., SEO, H., AND CORDIER, F. 2003. Automatic modeling of animatable virtual humans – a survey. In *4th International Conference on 3D Digital Imaging and Modeling (3DIM 2003)*, 2–11.
- MARQUARDT, D. W. 1963. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics* 11, 2, 431–441.
- MOHR, A., AND GLEICHER, M. 2003. Building efficient, accurate character skins from examples. *ACM Trans. Graph.* 22, 3, 562–568.
- SCHAEFER, S., AND YUKSEL, C. 2007. Example-based skeleton extraction. In *SGP '07: Proc. Eurographics Symposium on Geometry Processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 153–162.
- SEO, H., AND MAGNENAT-THALMANN, N. 2003. An automatic modeling of human bodies from sizing parameters. In *SI3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, 19–26.
- SUNDAR, H., SILVER, D., GAGVANI, N., AND DICKINSON, S. 2003. Skeleton based shape matching and retrieval. In *SMI '03: Proceedings of the Shape Modeling International 2003*, IEEE Computer Society, Washington, DC, USA, 130.
- TSCHIRREN, J., PALÁGYI, K., REINHARDT, J. M., HOFFMAN, E. A., AND SONKA, M. 2002. Segmentation, skeletonization, and branchpoint matching – a fully automated quantitative evaluation of human intrathoracic airway trees. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2002)*, Springer-Verlag, vol. 2489 of *Lecture Notes in Computer Science*, 12–19.
- WEBER, J. 2000. Run-time skin deformation. In *Proceedings of Game Developers Conference*.
- XIAN, X., SOON, S. H., FENG, T., LEWIS, J. P., AND FONG, N. 2006. A powell optimization approach for example-based skinning in a production animation environment. In *Computer Animation and Social Agents*, 141–150.
- XU, W., WANG, J., YIN, K., ZHOU, K., VAN DE PANNE, M., CHEN, F., AND GUO, B. 2009. Joint-aware manipulation of deformable models. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, ACM, New York, NY, USA, 1–9.
- ZELNIK-MANOR, L., AND PERONA, P. 2005. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, Cambridge, MA, 1601–1608.
- ZHANG, Y. 2009. User's guide for yall1: Your algorithms for 11 optimization. Tech. Rep. TR09-17, Rice University, May.