A Simple Framework for Natural Animation of Digitized Models

Edilson de Aguiar, Rhaleb Zayer, Christian Theobalt, Hans-Peter Seidel
Max-Planck-Institut für Informatik
Saarbrücken, Germany
{edeaguia, zayer, theobalt, hpseidel}@mpi-inf.mpg.de

Marcus Magnor
TU Braunschweig, Computer Graphics Lab
Braunschweig, Germany
magnor@cg.tu-bs.de

Abstract

We present a versatile, fast and simple framework to generate animations of scanned human characters from input optical motion capture data. Our method is purely mesh-based and requires only a minimum of manual interaction. The only manual step needed to create moving virtual people is the placement of a sparse set of correspondences between the input data and the mesh to be animated. The proposed algorithm implicitly generates realistic body deformations, and can easily transfer motions between human subjects of completely different shape and proportions. We feature a working prototype system that demonstrates that our method can generate convincing lifelike character animations directly from optical motion capture data.

1 Introduction

In recent years, photo-realistic computer-generated animations of humans have become the most important visual effect in motion pictures and computer games. In order to obtain an authentic virtual actor, it is of great importance that she mimics as closely as possible the motion of her real-world counterpart. Even the slightest unnaturalness would be instantaneously unmasked by the unforgiving eye of the viewer and the illusion of seeing a real person would be compromised.

It is thus no wonder that the number of working hours that animators spend in order to live up to these high requirements in visual quality is considerable. To generate virtual people, they make use of a well-established but often inflexible set of tools (see also Sect. 2) that makes a high amount of manual interaction unavoidable. First, the geometry of the human body is hand-crafted in a modeling software or obtained from a laser scan of a real individual. In a second step, a kinematic skeleton model is implanted into the body by means of, at best, a semi-automatic procedure. In order to couple the skeleton with the surface mesh, an appropriate representation of pose-dependent skin deformation has to be found. Finally, a description of body motion in terms of joint parameters of the skeleton is required. It can either be designed in a computer or learned from a real person by means of motion capture. Although the interplay of all these steps delivers animations of stunning naturalness, the whole process is very labor-intensive and does not easily allow for the interchange of animation descriptions between different virtual persons.

In this paper, we present a versatile, fast and simple approach to animate virtual characters. It uses a purely mesh-based animation paradigm that integrates into the traditional animation workflow. It can be used to realistically animate static meshes of arbitrary humans without relying on kinematic skeletons. Thus reducing the animator's effort to a great extent.

The presented framework produces realistic pose-dependent body deformations implicitly by means of a harmonic field interpolation. Furthermore, it solves the motion transfer problem, i.e. it enables the animator to interchange motions between persons of even widely different body proportions with no additional effort. Lastly, and most importantly, the method is able to give the animator instantaneous feedback when creating or modifying the animations, e.g. when specifying the small set of correspondences. Although we regard our system primarily as a tool for human animation, it can be applied in the same way to arbitrary moving subjects.

Figure 1. Subsequent frames generated by our system showing the female scan authentically performing a soccer kick. Note the realistic protrusion of the chest when she blocks the ball, as well as the original head motion.

The paper proceeds with a review of closely related work in Sect. 2. An overview of the approach is given in Sect. 3, and the nuts and bolts of our shape deformation method are described in Sect. 4. Sect. 5 demonstrates how characters can be directly animated using optical motion capture data as input. Finally, we show our animation results in Sect. 6 and conclude in Sect. 7.

2 Related Work

Our framework attempts to address many of the algorithmic subproblems arising in traditional human character animation by capitalizing on mesh deformation techniques presented in the field of geometry processing.

The first step in human character animation is the acquisition of a human body model comprising a surface mesh and an underlying animation skeleton [5]. Surface geometry can either be hand-crafted or scanned from a real person [3]. The skeleton model is either manually designed or inferred from input motion data [13]. It is also feasible to jointly create surface and skeleton models by fitting a template to body scans [4].

Mesh and skeleton have to be connected such that the surface deforms realistically with the body motion. A popular method serving this purpose is skinning [17]. It represents vertex displacements as weighted set of influences from adjacent joints. Weights can be hand-crafted or automatically inferred from examples [19]. Deformation models can also be created by interpolation between example scans [2]. Sand et al. [22] infer a skinning model by combining marker-based motion capture with a shape-from-silhouette method. Park et al. [20] presents a technique for capturing and animating surface deformations using a commercial motion capture system and approximately 350 markers.

The virtual human is awakened by specifying motion parameters for the joints in the skeleton. Common methods to generate such motion descriptions are key-framing [8], physics-based animation [10] or optimization-based creation of physically plausible movements [11]. The most authentic motion data can be acquired through optical motion capture systems [6]. Unfortunately, reusing motion capture data for subjects of different body proportions is not trivial, and requires computationally expensive motion editing [16] and motion retargetting techniques [12].

By extending ideas on mesh-based surface deformation we propose a versatile and simple framework as an alternative to the classic animation pipeline. As highly detailed 3D triangle meshes become more and more accessible, there has been an increasing interest in devising techniques which can work directly on these geometric representations without passing through intermediate pipelines such as the one mentioned above. In the mesh editing context, many approaches [1, 15, 18, 26] rely on the notion of differential coordinates to deform a mesh while preserving its geometric detail. This notion was extended to the volumetric setting in [27]. The main difference between these schemes lies in the way they propagate the deformation across the mesh. On the animation side, Sumner and Popovic [24] propose an approach that is similar in spirit but aims at a different goal. Using a full body correspondence between different synthetic models, their method can transfer the motion of one to the other.

Our system is most closely related to the SCAPE method [4]. The SCAPE model learns pose and shape variation across individuals from a database of body scans and can animate scanned human body geometry from motion capture by solving a nonlinear optimization problem. Our system addresses the animation problem by solving simple linear systems, hence delivering instantaneous results. By relying on the semantic similarities between characters it also provides an alternative solution to the retargetting problem.

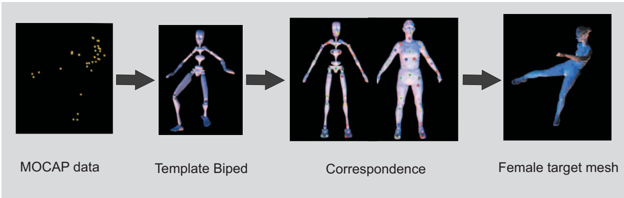

Figure 2. Illustration of our framework's pipeline.

3 Overview

It is the guiding idea behind the development of our framework to provide animators with a simple and fast method to directly apply captured motion to scanned human body models, Fig. 2. Thus, the input to our framework are motion data that have been measured from real individuals using optical motion estimation methods. The first processing step transforms these sequences of key body poses into a sequence of postures of a simple triangle mesh model, henceforth termed *template mesh*. Motion capture data can be straightforwardly transformed into a moving template mesh representation using standard animation software, Sect. 5. At the heart of our approach is an algorithm to transfer motion from the moving template mesh onto the scanned body mesh, henceforth termed the *target mesh*. We formulate the motion transfer problem as a deformation transfer problem, Sect. 4. To this end, a sparse set of triangle correspondences between the template and the target mesh needs to be specified and our automatic deformation interpolation method animates the target mesh. In Sect. 5 we show how our method is applied to directly generate character animations from motion capture data. We summarize our results in Sect. 6 and conclude in Sect. 7.

4 Mesh Deformation

The algorithmic core of our framework is a mesh deformation method that transfers the motion from the template mesh onto the target mesh. We regard motion transfer as a pure deformation interpolation problem, Sect. 4.1. This way, we put aside all difficulties relating to the dissimilarities between the template and the target, e.g. anatomical disparity (body proportions), and take advantage of their semantic similarities, e.g. the fact that both mesh representations have knees and elbows. For this purpose, the user is asked to specify a set of *correspondence triangles* between the two meshes. In practice, this means that the user marks a set of triangles on the template and assigns to each of them a corresponding triangle on the target. As described in Sect. 4.2, this can be interactively done using our prototype interface tool. We resort to this interactive step since there exists no viable automatic approach that can identify body segments on meshes standing in general poses. The first algorithmic challenge is to make the target mesh deform in the same way as the template by only considering the sparse set of representative triangle correspondences, Sect. 4.1. Furthermore, the generally large size of the high-resolution scans requires the use of fast numerical methods to allow for robust and efficient animation.

4.1 Deformation Interpolation

We formulate the deformation transfer problem as a deformation interpolation problem. The motion of the template mesh from its reference pose (e.g. Fig. 3) into another pose can be captured by the deformation applied to a set of marked triangles. A correct interpolation of this deformation applied over the corresponding triangles of the target mesh would bring it from its own reference pose (e.g. Fig. 3) into the template's pose. To this end, both reference poses are roughly aligned a priori. In the case of human animation, this deformation can be characterized as a simple rotation for each triangle and a translational degree of freedom.

If the per-triangle rotations are specified as simple 3×3 cosine matrices, the interpolation has to cope with $9 \times m$ interpolation points, where m is the number of triangle correspondences. This gives rise to an intricate problem, especially when the correlation between the components of each rotation matrix is taken into account. Nevertheless, there have been some successful approaches to propagate the effect of changing local frames on the whole mesh [18]. The quaternion representation, on the other hand, reduces the size of the range to $4 \times m$ interpolation points. Most of the existing methods for interpolating quaternion rotations rely on an intermediate function which guides the interpolation, as it has been shown in the case of the spherical or spline based interpolation. Such a function can be defined as distance function or heat kernels as in [25], or as a harmonic scalar field as proposed in [26]. Unfortunately, these methods do not apply to the current setting as they can only interpolate from a single set of starting points to a single set of ending points. Another breed of methods which allow for multiple point interpolation can be found in [7, 21]. While these techniques work well for locomotion, it is not clear how to extend them to our current general setup. In fact, it is difficult to find a function which can interpolate all the quaternion values and it is not clear how to assign specific rotations to all triangles in a mesh. Following an idea proposed in [26], we regard each component of a quaternion

$$Q = [w\ q_1\ q_2\ q_3] \quad (1)$$

as a scalar field defined over the entire mesh. Hence, given the values of these components at the marked triangles, we interpolate each scalar field independently. In order to guarantee a smooth interpolation we regard these scalar fields as harmonic fields defined over the mesh. The interpolation can then be performed efficiently by solving the Laplace equation over the whole mesh with constraints at the correspondence triangles:

$$\nabla^2 S \ = \ 0 \quad (2)$$

where S is a scalar field which is used here to represent alternatively each of the quaternion components w, q_1, q_2 and q_3. Once the rotational components are computed, we average the quaternion rotations of the vertices to obtain a quaternion rotation for each triangle. This way we establish a geometric transformation for each triangle of the target mesh M. However, this last step destroys its original connectivity and yields a new fragmented mesh M'. In order to recover the original geometry of the mesh while satisfying the new rotations, we have to solve the problem in a least square sense. In the following we sketch a simple way to setup this optimization problem that eases the implementation effort. The problem can be rephrased as finding a new tight mesh having the same topology as the original target mesh, such that its differential coordinates encode the same geometric detail as the ones of the fragmented mesh M'. This can be achieved by satisfying the following equation in terms of the coordinates x of M and u of M':

$$\nabla^2_M x \ = \ \nabla^2_{M'} u \, . \quad (3)$$

In order to carry out this discretization correctly the topological difference between both meshes should be addressed. Technically, the differential coordinates of the fragmented mesh are computed by deriving the Laplacian operator for the fragmented mesh and then applying it to its coordinates. This, in fact, yields a vector of size $3 \times nT$, where nT is the number of triangles. We sum the components of this vector according to the connectivity of the original mesh M. This yields a new vector $U_{reduced}$ of size nV, where nV is the number of vertices in M, and the discrete form of equation (3) reads as simple as

$$LX = U_{reduced} \, . \quad (4)$$

In this linear system the matrix L is the discrete Laplace operator. After solving this linear system individually for each coordinate direction x, y and z, the vertex positions are correctly reconstructed. In order to capture the local geometry of the mesh we use the geometric Laplacian, e.g. Desbrun et al. [9], which is more sensitive to the irregularities of the triangulation in comparison to the uniform or graph Laplacian. During the processing of an animation sequence, the differential operator matrix does not change. Furthermore, since it is symmetric positive definite we can perform a sparse Cholesky decomposition as preprocessing step and perform only back substitution for each frame. This enables us to compute novel poses of the target mesh at interactive rates for meshes of the order of 30 to 50 thousand triangles.

For simplifying the implementation of our framework, the deformation transfer approach is summarized as follows:

1. Calculate deformations for the selected triangles in the template mesh and extract the rotational components in quaternion representation.
2. Interpolate the associated quaternion representation over the whole target mesh M using equation (2).
3. Generate a fragmented mesh M' by applying the newly obtained quaternion to all triangles in M.
4. Compute the Laplacian of M' and apply it to its coordinates.
5. Reduce the length of the resulting 3D vector by summing up its components according to the connectivity of M.
6. Glue the fragmented mesh M' together by solving equation (4).

4.2 Correspondence Placement

As our method does not require any skeleton retargetting or full mesh correspondence, it is imperative that the choice of the sampling captures as much as possible of the geometric deformation.

In order to make the correspondence placement easy and intuitive we provide the prototype interface we used in our system, see Fig. 3. The tool allows access to both models simultaneously. An automatic alignment of the models is performed upon loading the mesh files. When a vertex is selected on one model, a corresponding vertex on the other model is highlighted. The artist can decide to keep this correspondence or improve it manually. For cylindrically shaped body parts we require the user to specify a single correspondence triangle and we mark additional triangles automatically by taking additional directions in the cross sectional plane and intersecting them with the mesh. For geometrically more complex body parts, such as the lap or the shoulders, correspondences are fully specified by the user.

Using our prototype interface the artist can verify instantaneously how the correspondences will influence the deformation transfer process. By loading a template mesh in a deformed pose, the target mesh can be deformed using the

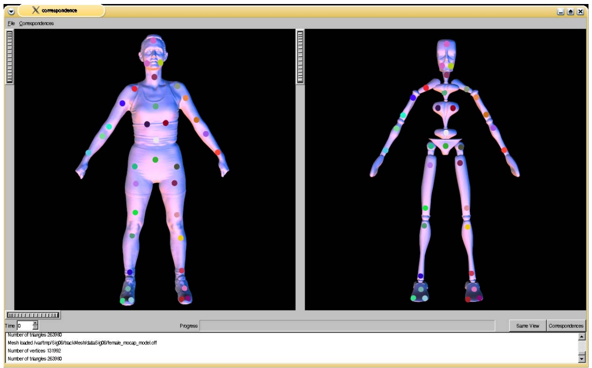

Figure 3. Prototype interface used in our system, featuring easy selection of correspondences and instantaneous feedback.

actual selected correspondences. This enables the artist to decide on-the-fly if the correspondences yield satisfactory results. Note that except from the positioning of the markers, our whole framework is fully automatic. The number of triangle correspondences used in our animations ranges from 140 to 220, half of which is automatically generated.

Furthermore, the placement of the markers directly affects the pose-dependent surface deformation of the target mesh. So the user does not have to tweak any weights as in the commonly used skinning methods. The principle here is simple: for having a sharp bend in the surface the correspondences should be placed close to either side of the joint, Fig. 4 (left). Increasing the distance of the markers from the joint allows for a softer bending, Fig. 4 (right).

4.3 Coping with Singularities

A limitation which is inherent to most systems based on rotation interpolation is the “candy-wrapper” collapse effect [19]. Our current approach is not immune to this problem and may suffer from its effects as well. In the following, we devise two simple techniques to tackle this problem.

In order to prevent the twisting collapse we need a simple way to predict when it happens. It occurs when some of the correspondence triangles undergo a rotation of 180 degrees. Conveniently, this can be easily detected by inspecting the first component of the quaternion representation. A null or a very small value indicates that such situation is likely to occur. The first solution to this problem is a simple workaround which basically consists of changing the reference pose of the template and the scan model and using an intermediate pose as a reference. To this end, a previous frame for the template and the target can be used as the reference pose. Another alternative would be to create an intermediate pose for the template mesh and deform the target into a similar pose. This intermediate pose can be used as the new reference configuration.

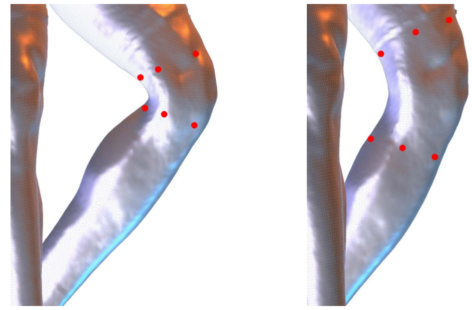

Figure 4. Influence of the markers’ placement on the deformation quality: marking correspondence triangles (red dots) close to an (anatomical) joint creates a sharp bend in the skin (left) while increasing the distance to the joint enables smoother bending (right).

The second solution would be to construct a shortest path between two points along which this problem occurs and interpolate the deformation along this line using the spherical quaternion interpolation commonly known as *SLERP*. This way, deformation can be interpolated to the triangles adjacent to the path. These new triangles will be used as constraints for the harmonic interpolation, Eq. (2).

5 Animating Digitized Models

By far the most authentic animation descriptions for moving virtual actors are obtained by measuring motion parameters of real people. The most established technique to achieve this purpose is optical motion capture [6]. Here, the body of a moving individual is equipped with optical markings at kinematically relevant body locations, e.g. around the joints. The person now performs in front of multiple cameras that reconstruct and track the 3D positions of the markings. From the 3D marker trajectories, a kinematic skeleton model of the person is generated. The motion is conveniently parameterized as rotational and translational parameters of the joints.

Motion capture systems provide the animator with motion descriptions of unequaled accuracy and naturalness. Even subtle details in motion patterns are faithfully captured. Furthermore, the motion data comes in the correct skeleton-based parameterization straight away, enabling the animator to straightforwardly map them onto a virtual character.

The high quality of captured motion data, however, comes at the expense of many inflexibilities in their application.

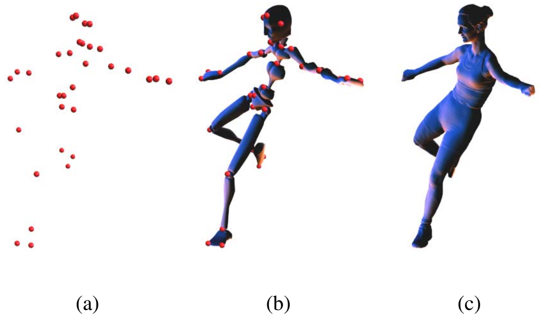

Figure 5. The set of input markers (a) are used to generate an intermediate biped model using any standard animation software (b). By applying our deformation technique the acquired motion is realistically transfered to the final human body scan (c).

Firstly, motion parameters cannot easily be reused with virtual persons that differ in skeletal proportions from the captured individual. To make this possible, computationally expensive motion retargetting algorithms have to be applied [12]. Secondly, motion capture systems only deliver a description of human motion in terms of interconnected rigid bodies. The non-rigid deformations of the skin and the soft-tissue surrounding the bones have to be manually modeled, e.g. by means of vertex skinning [17].

We now demonstrate that our animation paradigm can be straightforwardly applied to animate human body scans with motion capture data. Paradoxically, despite discarding the use of a kinematic animation skeleton it allows us to generate high-quality animations and to generate convincing surface deformations with just the same simple processing steps. By the same token, the motion retargetting problem is implicitly solved.

The steps that have to be taken to directly animate a mesh with the input data (Fig. 5a) are very simple and can be summarized in three sentences: First, using any standard animation software like 3D Studio Max^{TM}, the motion capture skeleton is transformed into a surface model in which the bones of the biped are represented as triangle meshes (Fig. 5b). Consequently, using our prototype interface static per-triangle correspondences between the triangulated biped and the scanned mesh are defined. Finally, our mesh deformation approach realistically moves and deforms the scanned mesh to accurately mimic the motion of the input model, and brings it to a correct global position (Fig. 5c).

We have applied our method to animate a male and a female mesh that have been generated with a Cyberware full-body scanner and that the company kindly provides for public use. Input motion capture data are taken from a database of motion files provided by Eyes, Japan Co. Ltd. Fig. 1 shows several frames of an animation in which we made the female model perform a soccer kick. The input is a motion capture file comprising of 90 key body poses. The actress realistically blocks the ball, kicks it and scores. Note that the animation nicely displays even subtle details like the protrusion of the chest during blocking. The skin deformation around the knees and the elbows is also authentically reproduced. Fig. 6 shows the male model performing boxing punches. Note that despite the fact that the input motions stem from persons with totally different anatomical dimensions, very natural animations free of retargetting artefacts (such as sliding feet) are generated. Our experiments confirm that our framework is a flexible and simple alternative to create character animations from arbitrary motion capture data.

Figure 6. Male model boxing, rendered with and without static surface textures. Note the realistic skin deformation of the animated scanned mesh.

6 Results and Discussion

To demonstrate the potential of our framework, we conducted several experiments with motion capture acquisition techniques. Due to their high resolution, we used the Cyberware models provided with their original surface colors in most of our experiments. In Fig. 1 and Fig. 6 the models faithfully reproduce the acquired performances of professional athletes.

The substantiated results and the accompanying video confirm that our method is capable of delivering visually convincing virtual character animation at a low interaction cost. The method is able to process large data sets in the order of 200 to 300 KΔ in just seconds. For smaller sets of 30 to 50 KΔ the results are generated at 2-5 frames per second. All the experiments were conducted on a single 3.2GHz notebook.

We see our method as an enhancement to the artist's traditional animation workflow. In order to evaluate its performance we conducted several experiments asking unexperienced users to animate a character using both our animation framework and a traditional animation software. A comparison of the resulting animations is shown in Fig. 7. This further confirms that our system is able to generate results comparable to professional animation packages, e.g. Character Studio ™, without requiring much user time or effort.

In the traditional animation pipeline, an unexperienced user needs many hours to correctly adjust the skinning weights. However in our system she was able to specify the correspondences quickly thanks to our prototype interface. After less than one hour she was able to produce the animation presented in Fig. 7. In fact, specifying correspondences is more intuitive to the user than working on building envelopes during the skinning process. In addition, correspondences can be tested instantaneously on our system, giving the user an adequate feedback.

Most recently, a multi-grid technique for efficient deformation of large meshes was presented [23] and a framework for performing constrained mesh deformation using gradient domain techniques has been developed in [14]. Both methods are conceptually related to our algorithm and could also be used for animating human models. However, none of the papers provides a complete integration of the surface deformation approach with a motion acquisition system, nor do they provide a comprehensive user interface to control the animation. On the other hand, we see potential use of the core of these methods within our framework for enhancing the animation quality and the speed of our system.

As for any novel technique our method still has some limitations. While our current system can handle motion capture data as input, it does not provide intuitive key-framing capabilities. For extreme deformation we note that there is generally some loss in volume due to the nature of our interpolation. We expect that using higher order differential operators or the volumetric approach proposed in [27] would reduce such artefacts although this might decrease the current numerical performance. Another limitation is that our system can not enforce hard constraints. Our method satisfies the deformation constraints in a least-square sense, while maintaining the smoothness and details of the original mesh which are encoded in the differential operators. Although it is not possible to explicitly enforce hard constraints, they can be implicitly enforced by increasing the number of correspondences associated with a marker. For instance we can ensure stable feet placement simply by marking a sufficient number of constraints on the feet. While allowing for an easy and intuitive control over the animation result, as mentioned in Sect. 4.2, a wrong placement of correspondences can lead to unsatisfactory animation results, in the same way as bad skinning weights do in the classical animation pipeline. However, during our experiments we could verify that bad correspondences can easily be detected and corrected using the instantaneous feedback provided by our prototype interface.

We nonetheless devised a powerful framework for animating virtual human characters. Since our method relies only on setting up and solving linear systems, the implementation and the reproduction of our results are straightforward.

7 Conclusion

Our animation framework aims at simplifying the traditional, not so straightforward acquisition-to-animation pipeline. The only manual interaction required is the selection of a small number of triangles to enforce the semantic correspondence between different models and guide the animation process. The proposed method is thus easy and intuitive to use and does not require any training. By means of the same efficient methodology our system simultaneously solves the animation, the surface deformation and the motion retargetting problem. As a direction for future work, we would like to combine our technique with an approach to learn per-time-step surface deformations from input video footage.

Acknowledgments: This work is supported by EC within FP6 under Grant 511568 with the acronym 3DTV.

References

[1] M. Alexa. Differential coordinates for local mesh morphing and deformation. *The Visual Computer*, 19(2-3):105–114, 2003.

[2] B. Allen, B. Curless, and Z. Popovic. Articulated body deformation from range scan data. *ACM Trans. Graph.*, 21(3):612–619, 2002.

[3] B. Allen, B. Curless, and Z. Popovic. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, 2003.

[4] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005.

[5] N. Badler, D. Metaxas, and N. Magnenat Thalmann. *Virtual Humans*. Morgan Kaufmann, 1999.

[6] B. Bodenheimer, C. Rose, S. Rosenthal, and J. Pella. The process of motion capture: Dealing with the data. In *Computer Animation and Simulation '97*, pages 3–18, Sept. 1997.

[7] S. R. Buss and J. P. Fillmore. Spherical averages and applications to spherical splines and interpolation. *ACM Trans. Graph.*, 20(2):95–126, 2001.

Figure 7. Direct comparison between a character animation generated by an animation software package (upper row) and our system (lower row). Our method is able to provide the same visual quality while requiring less effort and time from unexperienced users.

[8] J. Davis, M. Agrawala, E. Chuang, Z. Popovic, and D. Salesin. A sketching interface for articulated figure animation. In *Proc. of SCA '03*, pages 320–328. Eurographics Association, 2003.

[9] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum (Proc. Eurographics)*, 21(3):209–218, 2002.

[10] P. Faloutsos, M. van de Panne, and D. Terzopoulos. The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills. *Computers and Graphics*, 25(6):933–953, 2001.

[11] A. C. Fang and N. S. Pollard. Efficient synthesis of physically valid human motion. *ACM Trans. Graph.*, 22(3):417–426, 2003.

[12] M. Gleicher. Retargetting motion to new characters'. In *Proc. of ACM SIGGRAPH*, pages 33–42, 1998.

[13] L. Herda, P. Fua, R. Plänkers, R. Boulic, and D. Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *CA '00: Proc. of the Computer Animation*, page 77ff. IEEE Computer Society, 2000.

[14] J. Huang, X. Shi, X. Liu, K. Zhou, L.-Y. Wei, S.-H. Teng, H. Bao, B. Guo, and H.-Y. Shum. Subspace gradient domain mesh deformation. *ACM Trans. Graph.*, 25(3):1126–1134, 2006.

[15] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141, 2005.

[16] J. Lee and S. Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proc. of ACM SIGGRAPH*, pages 39–48, 1999.

[17] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proc. of ACM SIGGRAPH'00*, pages 165–172, New York, NY, USA, 2000.

[18] Y. Lipman, O. Sorkine, D. Levin, and D. Cohen-Or. Linear rotation-invariant coordinates for meshes. *ACM Trans. Graph.*, 24(3):479–487, 2005.

[19] A. Mohr and M. Gleicher. Building efficient, accurate character skins from examples. *ACM Trans. Graph.*, 22(3):562–568, 2003.

[20] S. I. Park and J. K. Hodgins. Capturing and animating skin deformation in human motion. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25(3), Aug. 2006.

[21] S. I. Park, H. J. Shin, and S. Y. Shin. On-line locomotion generation based on motion blending. In *Proc. of SCA '02*, pages 105–111. ACM Press, 2002.

[22] P. Sand, L. McMillan, and J. Popovic. Continuous capture of skin deformation. *ACM Trans. Graph.*, 22(3):578–586, 2003.

[23] L. Shi, Y. Yu, N. Bell, and W.-W. Feng. A fast multigrid algorithm for mesh deformation. *ACM Trans. Graph.*, 25(3):1108–1117, 2006.

[24] R. W. Sumner and J. Popovic. Deformation transfer for triangle meshes. *ACM Trans. Graph.*, 23(3):399–405, 2004.

[25] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum. Mesh editing with Poisson-based gradient field manipulation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 23(3):644–651, 2004.

[26] R. Zayer, C. Rössl, Z. Karni, and H.-P. Seidel. Harmonic guidance for surface deformation. In *Proc. of Eurographics 2005*, volume 24, pages 601–609, 2005.

[27] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H.-Y. Shum. Large mesh deformation using the volumetric graph laplacian. *ACM Trans. Graph.*, 24(3):496–503, 2005.