Rapid Animation of Laser-scanned Humans

Edilson de Aguiar* Christian Theobalt† Carsten Stoll‡ Hans-Peter Seidel§

MPI Informatik
Saarbrücken, Germany

Abstract

We present a simple and efficient approach to turn laser-scanned human geometry into a realistically moving virtual avatar. Instead of relying on the classical skeleton-based animation pipeline, our method uses a mesh-based Laplacian editing scheme to drive the motion of the scanned model. Our framework elegantly solves the motion retargeting problem and produces realistic non-rigid surface deformation with minimal user interaction. Realistic animations can easily be generated from a variety of input motion descriptions, which we exemplify by applying our method to both marker-free and marker-based motion capture data.

Keywords: Animation, virtual reality, motion capture, shape deformation.

Index Terms: I.3.7 [Computer Graphics]: Graphics and Realism—Animation, Virtual Reality; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Motion, Tracking

1 Introduction

The most important ingredients of many virtual worlds, being it computer games or virtual chat rooms to name just a few, are realistic virtual humans that populate them. In order to create lifelike avatars, several aspects of a person have to be convincingly modeled, including the shape of the person, her motion, but also the non-rigid deformation of the body surface. Nowadays, animators have a variety of tools at hand that enable them to get each of these aspects right. While it is still common to rely on the skills of artists to design realistic human geometry models, laser-scanning human shape represents a very fast and increasingly often used alternative. After designing the shape, an underlying skeleton structure that describes the kinematics of the character is fitted to the model in an interactive procedure. Subsequently, the influence of each bone on both rigid and non-rigid surface deformation needs to be specified, usually by handcrafting so-called "skinning envelopes". Stepping through this classical animation process can be rather time-consuming, and it becomes even prohibitive if many different virtual humans shall be quickly generated.

We thus propose a novel approach that streamlines the whole pipeline from laser-scanning to animation and that abandons the concept of a skeleton. It enables animators to quickly produce convincing animation results with minimal manual labor, while still allowing for control over the production process. Our method is based on a variant of Laplacian mesh editing [15]. It expects as input a geometry model (scanned or hand-crafted) given as triangle mesh, and a description of the motion this mesh shall perform. The input motion description needs to be transformed into a moving template model, i.e. a mesh or a point cloud. We will show later that this transformation step is trivial for many types of motion data, in particular captured motion. The animator maps the input motion onto the model by simply specifying a set of markers on the scanned human (*v-markers*) and corresponding markers on the template model generated from the input motion data (*t-markers*). Our method automatically generates convincingly skinned body surfaces and also allows for simple motion retargeting. Animations can be generated at near interactive frame rates. We demonstrate the efficiency of our algorithm by applying it to marker-based and marker-free motion capture data.

Our main contribution is the integration of a Laplacian mesh deformation scheme with motion capture systems to form an efficient and easy-to-use alternative to skeleton-based character animation. To this end, we have developed an intuitive animation tool providing full control over motion characteristics and deformation properties. We also show a simple way for extracting rotational constraints from point correspondences only.

The remainder of this paper is structured as follows: Sect. 2 reviews the most relevant related work. Sect. 3 details our mesh-based laplacian editing technique, which is responsible for the deformation of the target model at each frame of the animation. Sect. 4 shows possible applications of our method using both marker-based and marker-free motion capture data. Results and conclusion are presented in Sect. 5.

2 Related Work

The first step in traditional human character animation is the design of a model comprising of surface geometry and an underlying skeleton [3]. Surface geometry can either be hand-crafted or scanned from a real person [1]. The underlying skeleton structure is either manually designed or inferred from input motion data [7]. Geometry and skeleton need to be connected such that the surface deforms realistically with the body motion [10] .

The virtual human is animated by assigning motion parameters to the joints in the skeleton. By far, the most authentic motion representation can be acquired through marker-based [7] or marker-free motion capture systems [12]. Unfortunately, the reuse of motion capture data for different virtual models is not trivial, requiring computationally expensive motion editing [6] and motion retargeting techniques [18].

We have developed a novel simple and fast procedure that overcomes many limitations of the traditional animation pipeline by capitalizing on and extending ideas from mesh deformation. In [2] the authors propose a method to learn a parameterized deformable human model from a range of scans which could be used for animation.

Another category of approaches uses differential coordinates, see [15] for reviews on this subject. The potential of these methods for animation has already been stated in previous publications, however the focus always lay on deformation transfer between moving meshes. Using a complete set of correspondences between different synthetic models, [17] can transfer the motion of one model to the other. Following a similar line of thinking, [5] propose a mesh-based inverse kinematics framework based on pose examples with potential application to mesh animation.

In contrast to these approaches, our algorithm is a replacement

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

†e-mail: theobalt@mpi-inf.mpg.de

‡e-mail: stoll@mpi-inf.mpg.de

§e-mail: hpseidel@mpi-inf.mpg.de

for the classical way of animating humans from real-world motion data. It neither requires a dense set of surface correspondences, nor does it depend on a database of examples postures.

Recently, [14] presents a multi-grid technique for efficient deformation of large meshes and [8] presents a framework for performing constrained mesh deformation using gradient domain techniques. 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 does any of them provide a comprehensive user interface to control the animation. In contrast, we present a working prototype for rapid deformation-based animation from both marker-free and marker-based motion capture data. Our Laplacian mesh deformation method picks up and extends ideas presented in [16]. Since our approach only requires the solution of linear equation systems, animations can be generated at near interactive frame rates and an animation tool providing immediate visual feedback is feasible.

3 Mesh-based Animation Framework

3.1 Overview

Inputs to our method are a *target triangle mesh* of a human and a description of the motion that the mesh should perform. Following the classical animation pipeline, motion descriptions are often given in the form of rotation parameters of a kinematic skeleton. These descriptions can, for instance, be captured by marker-based or marker-free optical motion capture systems. To apply our framework, an input motion description has to be converted into a moving *template model*, which can either be a template triangle mesh or a template point cloud. In practice, it is often the simplest alternative to transform the input data into a moving template mesh, and we thus continue to use this term without loss of generality. In Sect. 4.1 and Sect. 4.2 we exemplify that it is straightforward to generate such a moving template from motion capture data that have been measured with a marker-based and a marker-free motion capture system, respectively. Note that we do not have any requirement to the template model. It is merely a tool helping us to set up point correspondences as we explain in the following.

It is our goal to transfer the motion of the template model to the target model. We formulate the motion transfer problem as a deformation transfer problem. After roughly aligning template and target models, a small set of corresponding marked vertices between the template and the target model is specified (Sect 3.2). Our automatic guided deformation interpolation technique is then used to animate the target human model efficiently (Sect 3.3).

3.2 Alignment and Correspondence Specification

In our algorithm, the motion of the template mesh from a reference pose into another pose is captured by the deformation of a small set of vertices marked under the guidance of the user. Applying these deformations to the corresponding markers in the target model would bring it from its own reference pose into the template's pose. For this purpose, we first roughly align template and target models in a given reference pose (Fig. 1). They are automatically aligned by applying a PCA-based alignment scheme to a reconstructed shape-from-silhouette volumetric model of the target mesh.

The user marks a set of K vertices on the template mesh, henceforth called *t-markers* $T_M = \{t_{Mk} | k \in \{0 \cdots K\}\}$, and assigns to each of them a corresponding vertex in the target model, thereby creating a set of *v-markers* $V_M = \{v_{Mk} | k \in \{0 \cdots K\}\}$.

Through placement of markers the characteristics of the motion and the surface skinning are defined, but also retargeting constraints can be set. We have developed a graphical user interface that assists the animator in controlling marker placement. A typical session consists of the following steps: First the user selects a vertex t_{Mk} in the template model. Since the target has been roughly aligned, the system proposes a corresponding closest v-marker, vertex v_{Mk}. Fortunately, we can compute deformed mesh poses at near interactive rates, and thus a new target pose is shown instantaneously after setting each pairwise correspondence. Due to the immediate visual feedback it is easy for the user to interactively modify correspondences. Except from this step, the whole pipeline is fully-automatic.

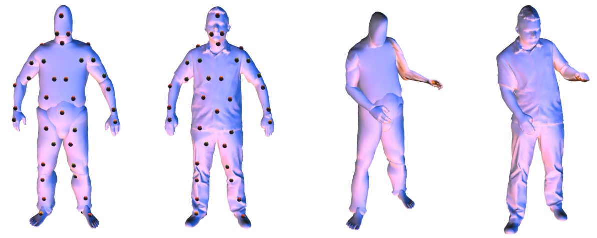

Figure 1: (a) Template (used for marker-free MoCap) and Target models at their reference poses. Dots represent corresponding marked points. (b) Models in a different pose. Note how the target human model is accurately deformed, mimicking the template model's pose.

Since the markers T_M and V_M will drive the deformation interpolation method, it is important that their choice captures as much as possible of the geometric deformation. The principle to place the markers is simple: they should be specified in areas where deformations are expected to happen, e.g. near anatomical joints. In addition, they can be specified in regions where the animator wants to enforce detailed deformation, e.g. on the torso, or explicit positional constraints. With the assistance of our interactive application, even unexperienced users quickly get a feeling of how to place markers. Each of the animations shown in Sect. 5 were generated by an unexperienced user in less than 15 minutes. Typically, between 35 to 65 markers are sufficient to create realistic animations.

3.3 Guided Deformation Interpolation

We employ a Laplacian mesh deformation scheme that jointly employs rotational and positional constraints on the markers similar to [16] to compute the sequence of poses for the avatar. The details of the target human model M are encoded in differential coordinates. The differential coordinates d of M are computed once at the beginning of the sequence by solving a linear system of the form $d = Lv$, where L is the discrete Laplace operator based on the cotangent-weights, and v is the vector of M's vertex coordinates [11]. Thereafter, our method performs the following three processing steps at each time step t of an animation:

(1) Local rotations for all markers V_M are estimated from the rotations of corresponding markers T_M between the template's reference pose and its pose at t.

(2) Local rotations are interpolated over the target mesh and the rotated differential coordinates of each vertex of M are determined.

(3) The model in the target pose is then reconstructed by solving the Laplace equation, subject to positional constraints derived from the positions of markers T_M at t.

Since the differential coordinates d are rotation-dependent [16], in step (1) we need to calculate the local rotations that should be applied to d. We propose a novel approach to derive these rotational constraints from moving points. The local rotation for each vertex v_{Mk} of M is calculated from the rotation of the corresponding t_{Mk} between reference pose and the pose at time t by means of a graph-based method. To this end, markers in T_M are considered as nodes in a graph, and edges between them are determined by constructing the minimal spanning tree [9]. For each marker t_{Mk}, we find the minimal rotation that makes the outgoing edges of t_{Mk} at the reference time matches its outgoing edges at time t (i.e. using the Jacobian) which is then converted to a quaternion $q_{t_{Mk}}$.

We want the markers V_M to perform the same rotations as their partners on the template, and thus state $q_{v_{Mk}} = q_{t_{Mk}}, k \in 0, \dots, K$. In step (2), we interpolate these rotations over M using the idea proposed in [19]. Each component of a quaternion $q = [w, q_1, q_2, q_3]$ is regarded as a scalar field defined over the entire mesh. A smooth interpolation is guaranteed by regarding these scalar fields as harmonic fields. The interpolation is performed efficiently by solving the Laplace equation $Lq = 0$ over the whole mesh with constraints at the marked vertices.

In step (3), we reconstruct the vertex positions v of M such that the mesh best approximates the rotated differential coordinates, as well as the positional constraints. This can be formulated as a least-squares problem of the form

$$\underset{v}{argmin}\{\|Lx - (q \cdot d \cdot \overline{q})\|^2 + \|Av - p\|^2\}. \tag{1}$$

which can be transformed into a linear system

$$(L^T L + A^T A)v = L^T (q \cdot d \cdot \overline{q}) + A^T p. \tag{2}$$

In Eq. 2 p is the vector of positional constraints of the form $v_j = p_j$, $j \in \{1, \dots, n\}$ specified for the K markers and possibly additional constraints set by the user. The matrix A is a diagonal matrix containing non-zero weights $A_{ij} = w_j$ only for constrained vertices j. Appropriate values for the weights w_j are found through experiments.

Since our framework employs a rotation interpolation technique, it can suffer from the ”candy-wrapper” collapse effect, mentioned in [13]. The twisting collapse effect happens when some of the markers t_{Mk} undergo a rotation of more than 180 degrees. It can be easily detected by verifying if the first component of q has a null or a small value. In this case, the reference pose of template and target model are changed using the previous pose and the calculation of the local rotations are performed again.

During the process of animating the target model the Laplacian matrix L does not change. Therefore, we are able to perform a sparse matrix decomposition, executing only back substitution for each frame. This enables us to compute target poses at a frame rate of 5 fps for models comprising of 10k to 30k triangles.

4 Sample Applications

We exemplify the efficiency and performance of our approach by generating animations from real-world motion data measured with both marker-free and marker-based motion capture systems.

4.1 Marker-based Animation

Nowadays, skeletal motion data acquired with a marker-based optical system are presumably amongst the most widely-used motion descriptions in animation production. It was thus one of our main motivations to develop a method that enables us to easily apply these data to high-quality surface models while bypassing the drawbacks of skeletal animation. Apart from being easy-to-use, the algorithm shall adhere to the same quality standard as the traditional way of animation.

Nearly all motion capture systems output a kinematic skeleton and a sequence of joint parameters. As stated earlier, we need to transform this kinematic representation into a moving template model. We straightforwardly achieve this by transforming the actual bone skeleton into a triangle mesh which can be automatically done by standard animation software like 3D Studio MAX^{TM}. Please note that we do not generate another surface model, and that there are no requirements at all concerning shape and connectivity of the template, apart from it containing moving vertices. Once the template has been generated, our guided deformation interpolation approach, Sect. 3.3, is applied to produce the animation.

We can even use the raw marker-trajectories output by the MoCap system as input. However, please note that the best positions of markers on the body for our purpose are different from the best marker positions for skeletal motion estimation. Obviously, most publicly available sequences have been captured with the latter application in mind, which makes it necessary to build a template prior to feeding them to our algorithm.

We have animated several laser-scanned subjects with motion files kindly provided to us by Eyes Japan Co. Ltd. We generated convincingly moving avatars, examples of which are shown in Fig. 2(a). Motion retargeting is feasible by appropriately placing constraints. On synthetic data, we could also verify that raw marker trajectories are a feasible input motion description.

4.2 Marker-free Animation

Instead of explicitly-placed markings on the body, marker-free systems employ natural image features to estimate motion parameters from video. Thus, they enable the person to wear comfortable everyday apparel during recording. Another advantage of this non-intrusive recording procedure is that the image data are available for further processing, e.g. for texture extraction.

We have build a silhouette-based marker-free motion capture system similar to [4] that employs a template body model comprising a segmented surface mesh and an underlying skeleton, Fig. 1, for the purpose of measuring motion parameters. Only eight video cameras are needed to faithfully capture motion data.

Since a mesh template is already used for tracking, we can employ our algorithm to straightforwardly map the captured motion to scans of arbitrary other persons. Please note that the specific shape of the template is a requirement of the tracking algorithm and not of the animation framework. Fig. 2(b,c) shows screenshots of animations that are obtained by mapping non-intrusively captured human performances to laser-scans of different subjects.

5 Results and Discussion

In both of our application scenarios, we animated body meshes of male and female subjects that were captured with a Cyberware^{TM} full-body scanner.

In the marker-based setting we generated results from 10 different motion sequences showing a variety of motions ranging from simple walking to soccer moves. The sequences were typically between 100 and 300 frames long. Fig. 2(a) shows several frames of different animations where the male model performs soccer moves. The target model realistically performs the motion while exhibiting lifelike non-rigid surface deformations. Note that the fact that the target human model has different dimensions compared with the recorded human subject is not a problem for our algorithm.

Marker-free animation examples are shown in Fig 2(b,c). Fig. 2(b) shows a comparison between actual input video frames and two models striking similar poses. It illustrates that our method can accurately transfer poses captured on video to the virtual characters. Fig. 2(c) shows some frames of a captured dancing sequence (330 frames) being mapped into a male model. More results are presented in the accompanying video.

The quality of the results confirms that we have developed a simple and efficient method to create moving virtual humans. Our algorithm is subject to a few limitations. For instance, during extreme

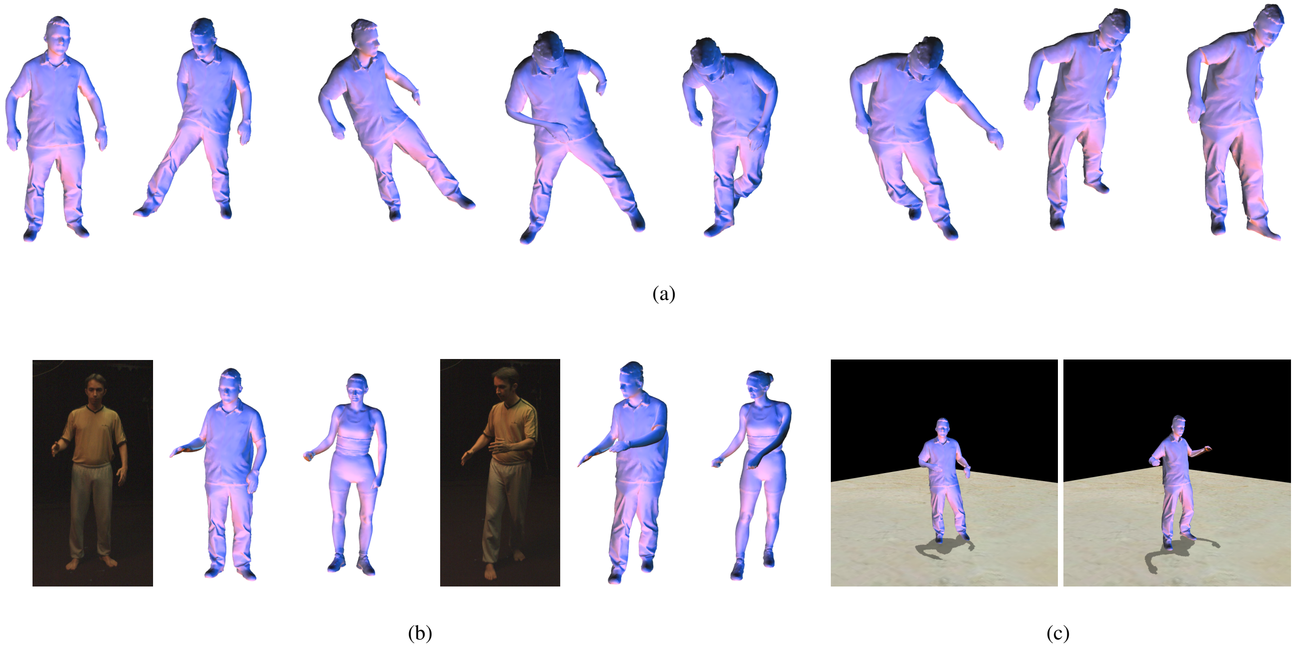

Figure 2: (a) Several frames of animations where the avatar performs soccer moves. The input motion was captured by means of a marker-based optical motion capture system. Note the lifelike non-rigid surface deformations automatically generated. (b) Two comparisons between an input video frame and two models striking the same pose. Although the models have different body dimension with respect to the real human subject, the poses are reconstructed accurately. (c) Two frames taken from an animation where a male model performs a captured dance.

deformations some loss in volume can occur. We believe that the incorporation of a volumetric constraint in Eq. 1 could overcome this limitation. However, by this means deformation modeling would turn into a nonlinear problem whose solution is numerically more involved [8]. Our method can generate animations at 5 fps for models comprising of 10k to 30k triangles. For VR applications, we can achieve better frame rates, as well as real-time performance, since the models used are usually smaller.

Nonetheless we presented a simple and fast scheme to create lifelike human characters that overcomes some limitations of the classical animation pipeline. Our system requires a minimum of manual interaction, only the placement of a small set of markers is required. The method is easy and intuitive to use, and simultaneously solves the animation, the surface deformation, and the motion retargeting problems.

Acknowledgements

This work is supported by EC within FP6 under Grant 511568 with the acronym 3DTV and AIM@SHAPE, a Network of Excelence project (506766) within EU's Sixth Framework Programme.

References

[1] 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.

[2] 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.

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

[4] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM Trans. Graph. (Proc. of SIGGRAPH'03)*, 22(3):569–577, 2003.

[5] K. G. Der, R. W. Sumner, and J. Popovic;. Inverse kinematics for reduced deformable models. *ACM Trans. Graph.*, 25(3):1174–1179, 2006.

[6] M. Gleicher. Motion editing with space-time constraints. In *Proc. of 1997 Symposium on Interactive 3D Graphics*, page 139ff, 1997.

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

[8] 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.

[9] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50, 1956.

[10] 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, 2000.

[11] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rössl, and H.-P. Seidel. Differential coordinates for interactive mesh editing. In *SMI 2004*, pages 181–190, 2004.

[12] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81(3):231–268, 2001.

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

[14] 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.

[15] O. Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4), 2006.

[16] C. Stoll, Z. Karni, C. Rössl, H. Yamauchi, and H.-P. Seidel. Template deformation for point cloud fitting. In *Symposium on Point-Based Graphics*, pages 27–35, 2006.

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

[18] S. Tak and H.-S. Ko. A physically-based motion retargeting filter. *ACM Trans. Graph.*, 24(1):98–117, 2005.

[19] 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.