

Learning Part-Based Models for Animation from Surface Motion Capture

Margara Tejera and Adrian Hilton
University of Surrey
Centre for Vision, Speech and Signal Processing
Guildford, United Kingdom
Email: {m.tejrapadilla, a.hilton}@surrey.ac.uk

Abstract—Surface motion capture (SurfCap) enables 3D reconstruction of human performance with detailed cloth and hair deformation. However, there is a lack of tools that allow flexible editing of SurfCap sequences. In this paper, we present a Laplacian editing technique that constrains the mesh deformation to plausible surface shapes learnt from a set of examples. A part-based representation of the mesh enables learning of surface deformation locally in the space of Laplacian coordinates, avoiding correlations between body parts while preserving surface details. This extends the range of animation with natural surface deformation beyond the whole-body poses present in the SurfCap data. We illustrate successful use of our tool on three different characters.

Keywords—surface performance capture; Laplacian deformation; animation; 3D video; part-based models;

I. INTRODUCTION

Multi-view video capture of actor performance has seen increased popularity over the past years due to its ability to capture clothing/hair detail and motion dynamics with both a naturalness and quality difficult to achieve by an animator [1], [2], [3], [4]. However, the lack of tools to create novel content from captured meshes hinders their introduction in the animation pipeline. A critical step for editing captured mesh sequences is the temporal alignment to obtain a mesh structure with surface correspondence over time referred to as surface motion capture (SurfCap) [2]. The input database for the work presented in this paper is the result of exploiting state-of-the-art non-sequential alignment approaches [5], [6] that obtain SurfCap sequences with a consistent mesh topology and vertex correspondence.

Animation from databases of SurfCap sequences has been demonstrated in the literature by the concatenation of captured segments [7], [8] and through re-sampling of video sequences [9], [10]. However, these approaches are limited to replay and do not offer the flexibility of conventional animation. Online interactive animation exploiting parameterisation of similar motions has been demonstrated recently [11], [12]. In this paper, we present a technique to generate novel poses of SurfCap meshes that maintains their inherent articulated structure and preserves the non-rigid deformation details of cloth and hair.

Techniques developed for the synthesis of novel articulated mesh poses generally consist of deforming a reference mesh within a subspace defined by a control skeleton [13].

This deformation consists in a combination of the effect of the transformations of the joints on the skin of the character. Example-based techniques enhance these results by accounting for non-linear natural skinning effects such as muscle bulging [14], [15], [16]. Pure surface-based techniques have also achieved mesh reposing by either learning from examples [17], [18] or by imposing skeletal constraints in order to maintain rigidity on the limbs [19]. Inspired by this family of techniques, the work presented in this paper synthesises novel poses within a Laplacian mesh editing framework enhanced with a space of surface deformation learnt from captured examples. Laplacian deformation provides a powerful tool for general mesh editing due to its ability to preserve local mesh details [20], [21]. The introduction of learnt constraints ensures that the mesh deformation is constrained to surface shapes in the space of mesh deformation observed in the captured data.

Previous work introduced global representation of the entire mesh shape in a single space [22]. This restricts the Laplacian mesh deformation to the space of example shapes which implicitly models the correlation between body parts. In this paper we introduce a local part-based representation which achieves considerably greater flexibility in mesh editing while preserving both the observed mesh structure and surface detail. Figure 1 illustrates the pipeline for our system: first, a set of example meshes are decomposed into mesh parts; second, a space of surface deformation is learnt from each part; and finally, local part-based editing is performed for a reference mesh. A hierarchical propagation of constraints allow seamless integration of the edited parts to form a complete surface model.

II. BACKGROUND

Production of animated mesh sequences is a mainstay for the film and game production industries. Ideally, an animator should count on applications that enable the generation of arbitrary shape deformations. Physics-based simulation engines produce dynamically correct animations that comply with the laws of Physics, however, they are computationally expensive and therefore not suitable for interactive frameworks. On the other hand, skeletal-controlled skinning algorithms such as Skeletal Subspace Deformation (SSD) [13], also known as Linear Blend Skinning, are ideal for

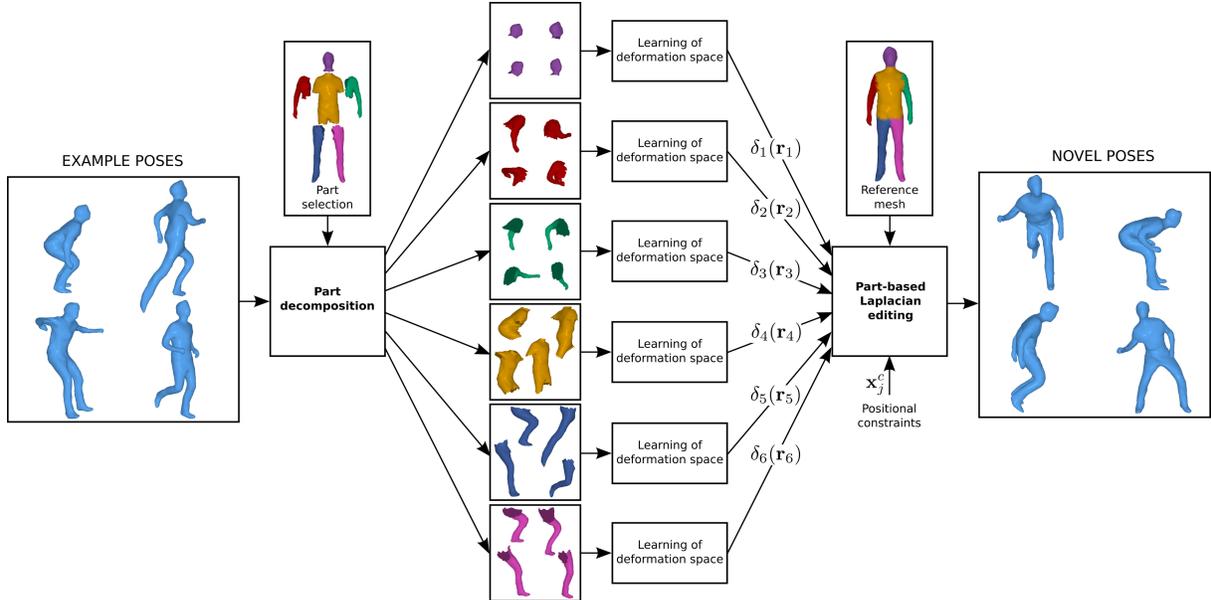


Figure 1. Overview of the animation method using learnt part-based models. First, a set of example poses are decomposed into mesh parts following the part selection chosen by the user. Second, a deformation space is learnt for each mesh part in the space of differential coordinates. Finally, a part-based Laplacian editing technique constrained by the space of deformations enables the creation of novel poses incorporating the soft positional constraints set by the user.

GPU implementation due to ease of parallelisation. These techniques provide real-time response but have two well-known artefacts: the candy wrapper effect due to twisting and mesh collapsing around the joints [14].

SSD uses a linear combination of skeletal joint transformations to control vertex positions. The quality of the deformation is dominated by the interpolation weights, which represent the influence of each joint on each vertex of the mesh. These weights can either be set manually or evaluated automatically: Baran and Popović [23] compute geometry-aware skinning weights from a single mesh by following the heat-equilibrium analogy over the skin of the character; Jacobson *et al.* [24] find the so-called bounded biharmonic weights as a result of minimising the Laplacian energy subject to bound constraints; and Kavan and Sorkine [25] propose weights such that the resulting deformation minimises a geometric elastic energy function discretised using a voxel grid enclosing the reference mesh.

The incorporation of example skeleton-mesh pairs has been a popular approach to account for deformations not present in the skeleton subspace defined by SSD. The mesh in the rest pose is animated in the example poses with traditional SSD and subsequently the deviations of these results from the corresponding example meshes are computed. Finally, given an unseen skeletal pose, a new displacement vector is predicted from the examples and the result of SSD skinning is corrected. Several techniques to perform this correction prediction can be found in the literature: Pose Space Deformation (PSD) [14] enables inclusion of

natural effects such as muscle bulging by interpolating vertex displacements directly; the EigenSkin technique [15] boosts the efficiency of PSD by applying Principal Component Analysis (PCA) and then interpolating directly in the space of coefficients; and Anguelov *et al.* [16] go a step further by learning both a pose deformation and a body shape model for human scans. The pose deformation model is built by finding the correlation between the rigid and non-rigid components of the triangles’ transformations and the shape model is represented by a linear transformation matrix whose components are learnt by performing PCA. This is a powerful approach since it enables the creation of realistic poses for a wide span of body shapes.

Several other mesh animation methods benefit from mimicking example meshes. They can be used for estimating skinning weights with a least-squares approach [26], [27], [28] or to introduce “context” to the deformations. Weber *et al.* [29] introduced log-quaternion interpolation to handle rotations correctly when combining the influence of joint transformations on the skin of the character. The deformations present in the examples which are not accounted for by this method are learnt in the space of *deformation gradients* and included in the final skinning result by finding local similarities between the target and the example poses. Wang *et al.* [30] follow a similar procedure but decompose deformation gradients displacements into rotational and scale/shear components and define both regression models to make predictions. Finally, examples can be used to study the correspondence between two sets of shape deformations

such as a naked body and a piece of garment, as demonstrated by Kim and Vendrovsky [31] with DrivenShape. This technique finds the blending weights needed to reproduce an unseen naked pose by the key example poses and then uses these same weights to synthesise the deformed garment. Residual displacements are applied to correct deformations for which linear interpolation does not suffice.

Relying on a skeleton simplifies the task of editing articulated meshes since it explicitly defines rigid segments and avoids distorting the articulated structure of the mesh. However, it restricts the deformations to valid articulated poses and gives limited freedom to the animator. In contrast, pure surface-based techniques can produce free-form deformation relying on a differential representation of the surface mesh, allowing the user to place positional constraints at a vertex level while preserving local details. The Laplacian operator extracts a differential representation of the mesh in the form of the so-called Laplacian or differential coordinates, which encode its local geometric properties: their direction approximates the direction of the normal of the triangles and their magnitude is proportional to the local mean curvature [20], [21]. Editing is performed by a least squares minimisation which tries to minimise the change of the Laplacian coordinates while fulfilling the constraints set by the user [32], [33]. This method has been widely used for mesh sculpting but the lack of information on the underlying structure of the mesh has hindered its use for animation. To overcome this, Huang *et al.* [19] formulate Laplacian mesh deformation as a non-linear least squares minimisation that can incorporate quasi-linear soft constraints and hard constraints. They solve this energy minimisation problem with the aid of a coarse control mesh to reduce slow convergence and demonstrate, among others, how a skeleton constraint can be imposed to maintain rigidity.

The main drawback of Laplacian deformation techniques is the linear nature of the differential coordinates which forces rotations to be handled explicitly. This problem was addressed by Lipman *et al.* [34] who introduced linear rotation-invariant coordinates allowing linear interpolation of meshes with correct rotations. Kircher and Garland [35] presented a differential surface representation invariant to both rotation and translation, based on computing first and second order differences of each vertex with respect to its neighbours. *Connection maps* that store these differences for each triangle enable motion processing operations to be performed with better results than vertex-based approaches.

Sumner *et al.* [17] express a set of example meshes by means of *deformation gradients* that represent the transformation that each triangle undergoes with respect to a reference mesh. By forcing the edited mesh to lie within the non-linear space of these deformation gradients and by Poisson stitching, they create novel poses with a valid anatomical structure. In previous work [22] we follow a similar methodology to perform space-time editing of mesh

sequences: a space of deformations that constrains a Laplacian editing is constructed from the observed whole-body poses. Creating a single space that considers the entire mesh encodes the correlation in the motion of different regions of the mesh, limiting the range of plausible output poses to be similar to the observed whole-body poses. In this paper, we present a local learnt part-based representation that removes these global correlations. This achieves greater animation flexibility by building a separate space of deformations for each mesh part while preserving the underlying structure of the entire mesh. Tena *et al.* [36] demonstrated the use of region-based models for face animation built in the space of vertex coordinates. In contrast, our approach is created in the space of differential coordinates exploiting their ability to encode local mesh details.

III. LAPLACIAN MESH EDITING

Mesh editing algorithms strive to achieve global changes to a mesh while maintaining its local characteristics, such as surface details. Differential mesh representations are suitable for this task due to their ability to encode the local geometric properties of the mesh. Laplacian coordinates, a particular case of these representations, are the basis for the Laplacian mesh editing framework.

Laplacian coordinates are extracted by means of the Laplacian operator, whose mathematical definition for a mesh of n vertices and m triangles is presented hereafter. The gradient of the triangles' basis functions ϕ_i yields a 3×3 matrix \mathbf{G}_j for each of the triangles [21]:

$$\begin{aligned} \mathbf{G}_j &= (\nabla\phi_1, \nabla\phi_2, \nabla\phi_3) \cdot (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)^\top & (1) \\ &= \begin{pmatrix} (\mathbf{p}_1 - \mathbf{p}_3)^\top \\ (\mathbf{p}_2 - \mathbf{p}_3)^\top \\ \mathbf{n}^\top \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{pmatrix}, & (2) \end{aligned}$$

where $\mathbf{p}_1, \mathbf{p}_2$ and \mathbf{p}_3 are the vertex positions and \mathbf{n} is the unit normal of the j^{th} triangle. In order to extract the gradients of all triangles of the mesh, a matrix \mathbf{G} of size $3m \times n$ can be constructed [37], such that:

$$\begin{pmatrix} \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_m \end{pmatrix} = \mathbf{G} \begin{pmatrix} \mathbf{p}_1^\top \\ \vdots \\ \mathbf{p}_n^\top \end{pmatrix}. \quad (3)$$

Let \mathbf{A} be a diagonal matrix of the areas of the triangles, then matrix $\mathbf{G}^\top \mathbf{A}$ represents the discrete divergence operator and $\mathbf{L} = \mathbf{G}^\top \mathbf{A} \mathbf{G}$ the Laplace-Beltrami operator [21]. This operator extracts the differential (or Laplacian) coordinates of a mesh from its absolute Cartesian coordinates: $\delta(\mathbf{x}) = \mathbf{L}\mathbf{x}$, $\delta(\mathbf{y}) = \mathbf{L}\mathbf{y}$ and $\delta(\mathbf{z}) = \mathbf{L}\mathbf{z}$. The direction of these coordinates approximate the direction of the normal of the triangles and their magnitude is proportional to the local mean curvature [20], [21]. Given a set of positional soft constraints, the reconstruction of the mesh that fulfils these constraints while maintaining its differential coordinates

and therefore preserves its local geometric features, can be performed with a least squares minimisation [20]:

$$\bar{\mathbf{x}} = \arg \min_x (\|\mathbf{L}\mathbf{x} - \delta(\mathbf{x}_o)\|^2 + \|\mathbf{W}_c(\mathbf{x} - \mathbf{x}_c)\|^2), \quad (4)$$

where \mathbf{x}_o are the coordinates of the original mesh and \mathbf{x}_c are the soft constraints on vertex locations given by the feature correspondence with a diagonal weight matrix \mathbf{W}_c .

This equation allows the reconstruction of a mesh by means of the Laplacian operator \mathbf{L} that, due to its linear nature, does not account for changes in local rotations that will most likely occur as a result of the editing operations. To overcome this limitation, the differential coordinates are explicitly updated following an iterative approach [32], [38] performed for each step of the minimisation: we reconstruct the mesh from the differential coordinates, extract the rotations of the triangles in the local frame defined by two of their edges and their normal, and finally rotate the differential coordinates accordingly.

IV. PART-BASED LEARNT LAPLACIAN EDITING

Surface motion capture produces mesh sequences with dynamic surface deformation details, such as the motion of clothing during actor performance. The traditional Laplacian mesh editing framework presented in Section III enables preservation of the local geometric characteristics of a mesh but is unable to provide non-rigid surface deformations that would naturally appear for certain mesh configurations. For instance, the change in shape of a shirt that occurs for different arm poses. Moreover, it does not contain any semantic information on the underlying structure of the mesh, which is particularly important for articulated meshes.

Building on previous work [22], we propose a part-based learning scheme that captures the non-rigid deformations and underlying structure present in SurfCap sequences from a set of example poses. Within a hierarchical Laplacian editing framework, this approach can produce mesh edits that reproduce the shape structure and surface details of the original mesh sequences.

A. Learning part-based models

Creating a space of deformations that considers the mesh as a whole will capture the correlation between the different regions of the mesh. For instance, as illustrated in Figure 2 for our database of human-like meshes, if the user edits one of the legs towards a new position present in the example meshes, the system would drive the other towards the configuration that corresponds to that leg position. To overcome this limitation we propose to decompose the mesh into parts that can be edited independently with the aid of their own deformation space.

Part decomposition is performed by dividing the mesh into a set of N overlapping submeshes of n_j vertices, as exemplified in Figure 3. For the particular case of SurfCap meshes of actor performance, the most intuitive way of decomposing

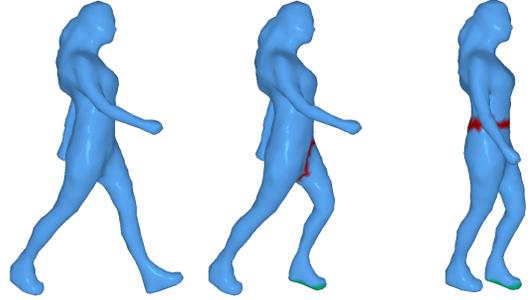


Figure 2. Comparison of Laplacian editing with global and part-based basis. The green area on the left foot of the character represents the handle and the red fixed vertices. The remainder of the mesh is unconstrained. Left: original mesh. Centre: result of dragging the left foot backwards using a local basis for the left leg. Right: same edit performed using a global basis. Note how the deformer has dragged the right leg forward and the right arm backwards, both unconstrained, due to the effect of the basis.

the mesh is by following its underlying articulated structure. As shown in Figure 5, we define six parts: one for each arm and leg, one for the head and one for the torso. Overlapping regions account for the interrelation between mesh parts. For instance, an edit in a leg will affect vertices on the torso close to the leg, and vice versa.

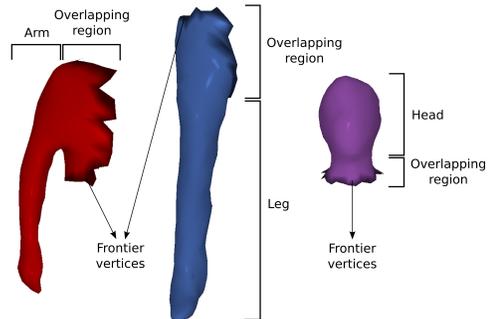


Figure 3. Examples of part selection. Each part of the limbs/head is composed of the vertices of the body part that it represents plus a region of vertices that overlaps with the torso.

Figure 4 illustrates the process of learning a deformation space for a mesh part. Given a set of F example meshes $\{M_i\}_{i=1}^F$ decomposed in N parts, a set of example submeshes $\{M_{ij}\}_{j=1}^N$ are created for each of the parts. For every part j the differential coordinates $\delta_{ij}(\mathbf{x}_{ij})$, $\delta_{ij}(\mathbf{y}_{ij})$ and $\delta_{ij}(\mathbf{z}_{ij})$ are extracted by means of the Laplacian operator \mathbf{L}_j , forming a data matrix \mathbf{M}_j of size $F \times 3n_j$:

$$\mathbf{M}_j = \begin{pmatrix} \delta_{1j}^\top(\mathbf{x}_{1j}) & \delta_{1j}^\top(\mathbf{y}_{1j}) & \delta_{1j}^\top(\mathbf{z}_{1j}) \\ \delta_{2j}^\top(\mathbf{x}_{2j}) & \delta_{2j}^\top(\mathbf{y}_{2j}) & \delta_{2j}^\top(\mathbf{z}_{2j}) \\ \vdots & \vdots & \vdots \\ \delta_{Fj}^\top(\mathbf{x}_{Fj}) & \delta_{Fj}^\top(\mathbf{y}_{Fj}) & \delta_{Fj}^\top(\mathbf{z}_{Fj}) \end{pmatrix} \quad (5)$$

Matrix \mathbf{M}_j represents a set of surface deformation ex-

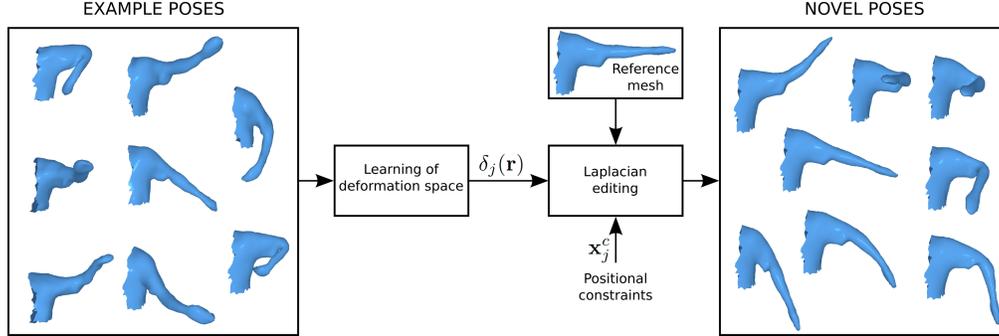


Figure 4. Process of learning a deformation space from a set of examples for a mesh part. Novel unseen poses can be synthesised by a Laplacian editing constrained to the space of observed deformations.

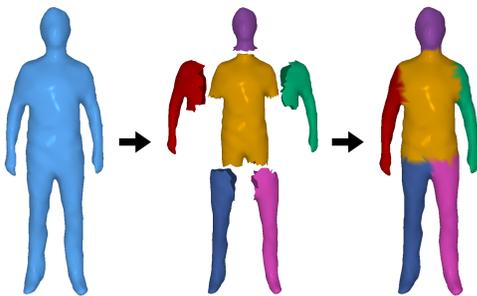


Figure 5. Part division. Left: reference mesh. Centre: submeshes resulting from the manual body part selection. Right: full body mesh built as a combination of the part submeshes.

amples that can be represented by an orthogonal basis \mathbf{V}_j extracted from the SVD decomposition of matrix $\mathbf{M}_j = \mathbf{U}_j \mathbf{D}_j \mathbf{V}_j^\top$. Basis \mathbf{V}_j of size $3n_j \times F$ contains a basis vector \mathbf{e}_{jk} in each column k , $1 \leq k \leq F$, and matrix \mathbf{D}_j has the singular values of \mathbf{M}_j in its diagonal. To perform dimensionality reduction and remove the high-frequency noise that may be present in the data, we keep the first l_j basis vectors that account for 95% of the energy of the data, defining a space of surface deformation for part j :

$$\delta_j(\mathbf{r}_j) = \bar{\delta}_j + \sum_{k=1}^{l_j} r_{jk} \mathbf{e}_{jk} = \bar{\delta}_j + \mathbf{E}_j \mathbf{r}_j, \quad (6)$$

Where r_{jk} are scalar weights for each basis vector \mathbf{e}_{jk} , \mathbf{r}_j is an l_j -dimensional weight vector and \mathbf{E}_j is an $3n_j \times l_j$ matrix whose columns are the first l_j basis vectors of length $3n_j$.

B. Part editing and mesh recombination

From equation 4 we define a set of N Laplacian deformer constrained by the space of surface deformations presented in the previous section [22]:

$$\tilde{\mathbf{r}}_j, \tilde{\mathbf{x}}_j = \arg \min_{\mathbf{r}_j, \mathbf{x}_j} (\|\mathbf{L}_j \mathbf{x}_j - \delta_j(\mathbf{r}_j)\|^2 + \|\mathbf{W}_j^c (\mathbf{x}_j - \mathbf{x}_j^c)\|^2) \quad (7)$$

Where $\tilde{\mathbf{x}}_j$ are the absolute Cartesian coordinates of the reconstructed part and $\tilde{\mathbf{r}}_j$ represents the point in the space of surface deformation δ_j where the resulting mesh lies. As demonstrated in Figure 4, constraining the Laplacian deformation allows the creation of novel part poses that preserve the surface details and underlying structure of the original examples.

Mesh parts are all members of a higher level structure defined by the full mesh and are interrelated by the sharing of vertices. Therefore, the edit of a part cannot be handled as an isolated process. In order to account for this interrelation, we propose to represent the mesh by a hierarchical structure defined by vertex sharing. A tree of parts whose root is selected by the user can be built by analysing the overlapping regions of the parts. In the particular case of our SurfCap database, we build a simple tree with the torso acting as the root, as depicted in Figure 6. When a mesh edit is performed on any part of the tree, the deformation is propagated to all parts following the hierarchical structure.

Propagation from parent to child parts of the hierarchy: When a part edit is performed, all parts in the lower level of the hierarchy –child parts– must be positioned accordingly to ensure the integrity of the full mesh. Let $\{I_{jp}\}$ be the set of vertex indices of the frontier vertices for child part j which are shared with parent part p . When part p is edited, for each child part j the position of vertices $\{I_{jp}\}$ are extracted from the new configuration of part p to form a vector \mathbf{x}^c that will act as positional soft constraints for the deformation of each part j . This process is performed recursively until reaching the leaves of the tree.

Mesh blending with the upper level of the hierarchy: In order to avoid abrupt changes of the mesh geometry, the edited part j is smoothly blended with its parent p by setting the frontier vertices I_{jp} as anchors for the editing, as illustrated in Figure 7 for the the right arm and the torso. Since the least squares minimisation minimises the change of the differential coordinates, this ensures that the mesh in the proximity of the frontier vertices will deform smoothly. Moreover, to account for the cases of extreme deformations

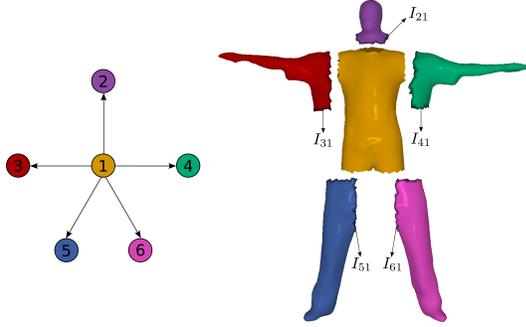


Figure 6. Tree structure for a six-part decomposition of a human-like mesh. For this example, the torso deformation will be propagated to its child parts (head, arms and legs) using the frontier vertices that they share with the torso, I_{jp} , as positional constraints for the deformation.

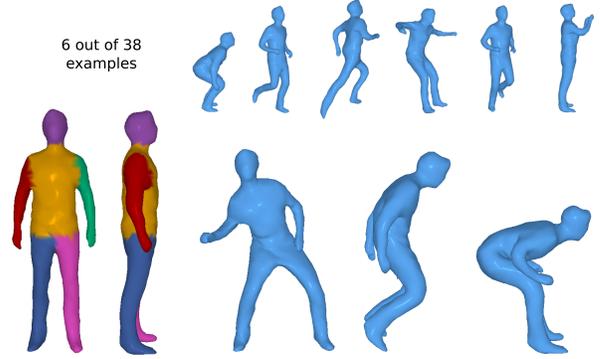


Figure 9. Results for character “Dan” showing novel poses not present in the original SurfCap data used for training. 2667 vertices. 5330 triangles. 38 examples.

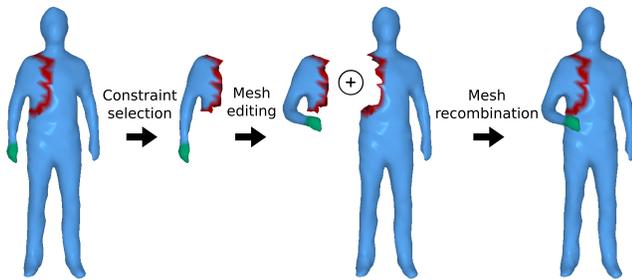


Figure 7. Part editing and mesh recombination. In this example, the frontier vertices (red) of the arm shared with the torso are set as anchors for the deformation. This ensures that the recombination of the edited arm with the remainder of the mesh is smooth.

when the positional soft constraints are not met exactly, the final position of the frontier vertices is computed by linearly interpolating their position in the edited part j and in the parent part p .

V. RESULTS

Figure 8 shows an editing session with the intermediate results and constraints until reaching a final pose for character “Dan”. Figures 9, 10 and 11 illustrate the performance of our editing algorithm on three different characters. Videos of these results can be found in the supplementary video¹. These example edits demonstrate that the part-based approach allows creation of novel poses not observed in the captured data with plausible surface deformation. It is important to remark that the range of motion present in the original SurfCap sequences available for each character varies and this affects the richness of the deformations that can be achieved with our system. Example poses capturing as much variety of deformation as possible were manually selected from these sequences. Note we do not learn the shape space from all available frames as this may bias the representation towards common poses and omit

¹http://cvssp.org/Personal/MargaraTejera/Tejera_3DV2013.avi

representation of extreme poses which add to the diversity of the space.

Our editing interface achieves interactive rates with an editing time (deformation plus recombination) of 0.3 seconds for a mesh of 1606 vertices on an Intel Core i5-2500 CPU at 3.30GHz.

VI. CONCLUSION

We have presented a novel part-based Laplacian mesh editing approach that is constrained by a set of learnt deformation models to produce plausible editing results. Mesh editing methods such as linear rotational coordinates [34] and implicit rotation optimisation [33] that overcome the lack of rotation representation of the differential coordinates could in theory be constrained likewise. However, they introduce other limitations: the former cannot deal with pure translations and the latter is based on a linearisation of rotations and is only valid for angles up to $\pi/2$ [21]. The naive iterative rotational update used in this paper has been sufficient for the editing tasks performed with our interface, further investigation on the topic will be part of future work.

Spaces of surface deformation are learnt for each mesh part from a set of examples. This removes the global correlations between mesh regions and enables the capture of local surface details. Part-based learning of surface deformation allows local manipulation of the mesh and achieves greater animation flexibility than global learning approaches which restrict output meshes to the space of observed whole-body poses. The smoothness of the complete surface model is ensured by overlapping mesh regions and by a hierarchical propagation of constraints.

Our approach has been demonstrated on a database of SurfCap meshes of actor performance for a specific six-part decomposition. However, any hierarchical decomposition can be used according to the level of manipulation desired by the animator. Automatic algorithms that identify mesh regions with correlated motion could also be incorporated.

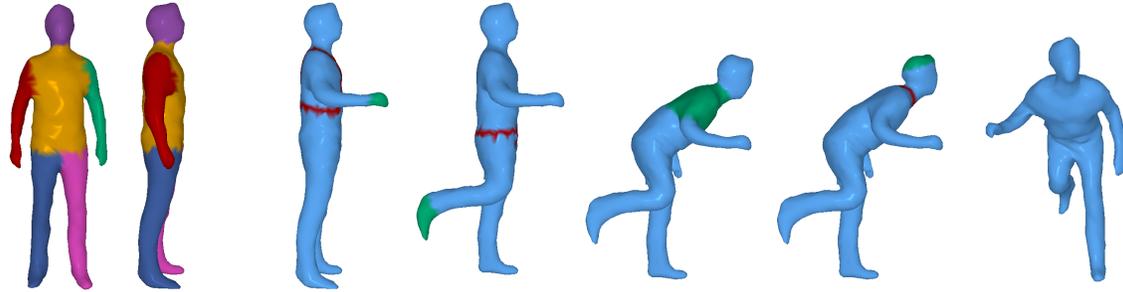


Figure 8. Editing session. From a regular standing pose (first from the left), the character is driven towards a bending motion with the right arm and leg lifted (first from the right). Green and red areas on the blue meshes represent the handle and anchor regions, respectively.

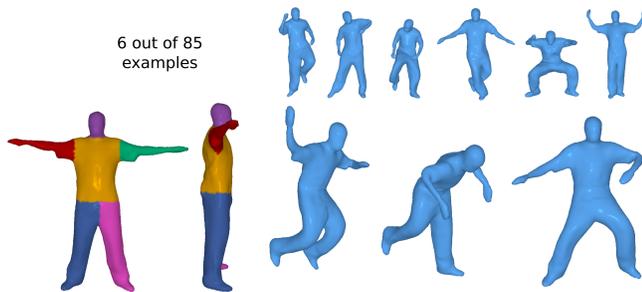


Figure 10. Results for character “JP” showing novel poses not present in the original SurfCap data used for training. 5580 vertices. 11156 triangles. 85 examples.

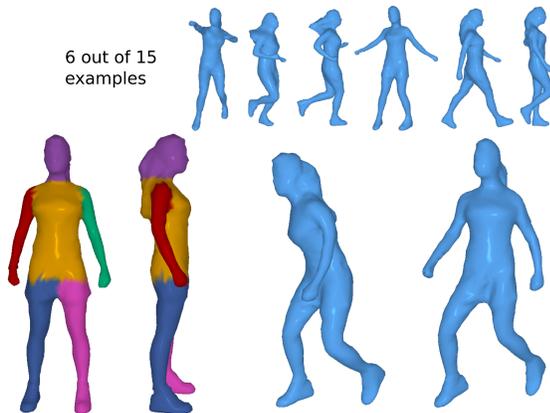


Figure 11. Results for character “Roxanne” showing novel poses not present in the original SurfCap data used for training. 2475 vertices. 4950 triangles. 15 examples.

ACKNOWLEDGMENT

This work was supported by EPSRC platform Grant (EP/F02827X/1) and EU project SCENE (ICT-287693).

REFERENCES

[1] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality video view interpolation using a layered representation,” 2004, pp. 600–608. [1](#)

[2] J. Starck and A. Hilton, “Surface capture for performance-based animation,” *IEEE Computer Graphics and Applications*, vol. 27, no. 3, pp. 21–31, 2007. [1](#)

[3] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, “Performance capture from sparse multi-view video,” *ACM SIGGRAPH*, vol. 27, no. 3, 2008. [1](#)

[4] D. Vlastic, I. Baran, W. Matusik, and J. Popović, “Articulated mesh animation from multi-view silhouettes,” in *Proceedings of ACM SIGGRAPH*, 2008. [1](#)

[5] P. Huang, C. Budd, and A. Hilton, “Global temporal registration of multiple non-rigid surface sequences,” in *CVPR*, 2011. [1](#)

[6] C. Budd, P. Huang, M. Kludiny, and A. Hilton, “Global non-rigid alignment of surface sequences,” *International Journal of Computer Vision*, pp. 1–15, 2012. [1](#)

[7] J. Starck, G. Miller, and A. Hilton, “Video-based character animation,” in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005, pp. 49–58. [1](#)

[8] P. Huang, A. Hilton, and J. Starck, “Human motion synthesis from 3D video,” in *CVPR*, 2009. [1](#)

[9] M. Flagg, A. Nakazawa, Q. Zhang, S.-B. Kang, Y. Ryu, I. Essa, and J. Rehg, “Human Video Textures,” in *ACM Symposium on Interactive 3D Graphics*, 2009. [1](#)

[10] F. Xu, Y. Liu, C. Stoll, J. Tompkin, G. Bharaj, Q. Dai, H.-P. Seidel, J. Kautz, and C. Theobalt, “Video-based characters - creating new human performances from a multi-view video database,” *Proceedings of ACM SIGGRAPH*, 2011. [1](#)

[11] D. Casas, M. Tejera, J.-Y. Guillemaut, and A. Hilton, “4D parametric motion graphs for interactive animation,” in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '12. New York, NY, USA: ACM, 2012, pp. 103–110. [1](#)

[12] —, “Interactive animation of 4D performance capture,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 5, pp. 762–773, 2013. [1](#)

- [13] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann, "Joint-dependent local deformations for hand animation and object grasping," in *Proceedings on Graphics interface '88*. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 1988, pp. 26–33. [1](#)
- [14] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 165–172. [1](#), [2](#)
- [15] P. G. Kry, D. L. James, and D. K. Pai, "EigenSkin: real time large deformation character skinning in hardware," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ser. SCA '02. New York, NY, USA: ACM, 2002, pp. 153–159. [1](#), [2](#)
- [16] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: shape completion and animation of people," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 408–416, Jul. 2005. [1](#), [2](#)
- [17] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović, "Mesh-based inverse kinematics," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*. New York, NY, USA: ACM, 2005, pp. 488–495. [1](#), [3](#)
- [18] K. G. Der, R. W. Sumner, and J. Popović, "Inverse kinematics for reduced deformable models," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*. New York, NY, USA: ACM, 2006, pp. 1174–1179. [1](#)
- [19] 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," in *ACM SIGGRAPH 2006 Papers*, ser. SIGGRAPH '06. New York, NY, USA: ACM, 2006, pp. 1126–1134. [1](#), [3](#)
- [20] O. Sorkine, "Differential representations for mesh processing," *Computer Graphics Forum*, vol. 25, no. 4, pp. 789–807, Dec. 2006. [1](#), [3](#), [4](#)
- [21] M. Botsch and O. Sorkine, "On linear variational surface deformation methods," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 1, pp. 213–230, 2008. [1](#), [3](#), [6](#)
- [22] M. Tejera and A. Hilton, "Space-time editing of 3D video sequences," in *Proceedings of the 2011 Conference for Visual Media Production*, ser. CVMP '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 148–157. [1](#), [3](#), [4](#), [5](#)
- [23] I. Baran and J. Popović, "Automatic rigging and animation of 3D characters," *ACM Trans. Graph.*, vol. 26, Jul. 2007. [2](#)
- [24] A. Jacobson, I. Baran, J. Popović, and O. Sorkine, "Bounded biharmonic weights for real-time deformation," in *ACM SIGGRAPH 2011 papers*, ser. SIGGRAPH '11. New York, NY, USA: ACM, 2011. [2](#)
- [25] L. Kavan and O. Sorkine, "Elasticity-inspired deformer for character articulation," *ACM Trans. Graph.*, vol. 31, no. 6, Nov. 2012. [2](#)
- [26] X. C. Wang and C. Phillips, "Multi-weight enveloping: least-squares approximation techniques for skin animation," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ser. SCA '02. New York, NY, USA: ACM, 2002, pp. 129–138. [2](#)
- [27] A. Mohr and M. Gleicher, "Building efficient, accurate character skins from examples," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03, vol. 22, no. 3. New York, NY, USA: ACM, 2003, pp. 562–568. [2](#)
- [28] D. L. James and C. D. Twigg, "Skinning mesh animations," in *ACM Trans. Graph.*, 2005, pp. 399–407. [2](#)
- [29] O. Weber, O. Sorkine, Y. Lipman, and C. Gotsman, "Context-aware skeletal shape deformation," *Computer Graphics Forum (Proceedings of EUROGRAPHICS)*, vol. 26, no. 3, pp. 265–273, 2007. [2](#)
- [30] R. Y. Wang, K. Pulli, and J. Popović, "Real-time enveloping with rotational regression," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007. [2](#)
- [31] T. Y. Kim and E. Vendrovsky, "DrivenShape: a data-driven approach for shape deformation," in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '08. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2008, pp. 49–55. [3](#)
- [32] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rössl, and H.-P. Seidel, "Differential coordinates for interactive mesh editing," in *Proceedings of Shape Modeling International*. IEEE Computer Society Press, 2004, pp. 181–190. [3](#), [4](#)
- [33] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proceedings of the EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*. ACM Press, 2004, pp. 179–188. [3](#), [6](#)
- [34] Y. Lipman, O. Sorkine, D. Levin, and D. Cohen-Or, "Linear rotation-invariant coordinates for meshes," *ACM Trans. Graph.*, vol. 24, pp. 479–487, July 2005. [3](#), [6](#)
- [35] S. Kircher and M. Garland, "Free-form motion processing," *ACM Trans. Graph.*, vol. 27, no. 2, pp. 12:1–12:13, May 2008. [3](#)
- [36] J. R. Tena, F. De la Torre, and I. Matthews, "Interactive region-based linear 3D face models," in *ACM SIGGRAPH 2011 papers*, ser. SIGGRAPH '11. New York, NY, USA: ACM, 2011, pp. 76:1–76:10. [3](#)
- [37] M. Botsch, R. W. Sumner, M. Pauly, and M. Gross, "Deformation transfer for detail-preserving surface editing," in *Proc. Vision, Modeling, and Visualization*, 2006, pp. 357–364. [3](#)
- [38] C. Stoll, E. de Aguiar, C. Theobalt, and H.-P. Seidel, "A volumetric approach to interactive shape editing," Max-Planck-Institut für Informatik, Tech. Rep., June 2007. [4](#)