Wrinkle Flow for Compact Representation of Predefined Clothing Animation

Byung-Uck Kim Farzam Farbiz
Institute for Infocomm Research, A*STAR, Singapore *

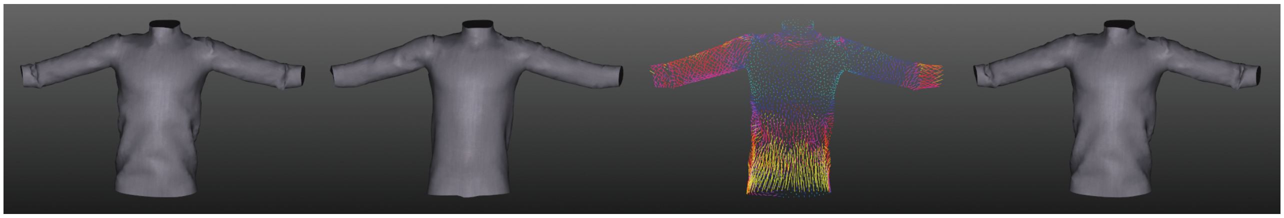

Figure 1: *From left to right, (a) Original animation (Ground truth), (b) Skinned clothing animation, (c) Wrinkle Flow, (d) Our method.*

1 Introduction

Animations of a clothed character play a key role in enriching visual realism in various CG applications. However, the challenge remains in finding a compact representation of predefined clothing animations while preserving fine-scale details such as wrinkles and folds. In this work, we present a method to decouple clothing animation into a skinning and a non-skinning component. The skinning component controls the general look of a clothing mesh, which is derived from the underlying kinematic character animation; the non-skinning component mainly contributes to the formation of wrinkles, which is further reduced into a lower dimensional subspace to support a compact representation as depicted in Kry *et al.*'s [2002] work. From a practical viewpoint, our method offers a simple, fast, and effective way to compactly encode predefined clothing animations with fine details.

2 Our Approach

Input data Clothing animations on a virtual character can be prerecorded through cloth simulation and/or artist's direct manipulation. In our experiment, we employ a commercial cloth simulation package [1] to animate a clothing mesh on top of a skinned character animation from the CMU motion capture database [2]. Such predefined clothing animation with m frames can be represented with the matrix $\mathbf{X} = [x^1, x^2, \dots, x^m]$, where $x^i_j \in \mathbb{R}^3$ indicates j-th vertex position at i-th frame and $x^i \in \mathbb{R}^{3n}$ for n vertices (thus, $size(\mathbf{X}) \approx O(3n \times m)$).

Skinning component with Deformation Transfer In general, a clothing mesh conforms to the underlying character pose and shape, and this is particularly pertinent for tight-fitting garments such as shirts and pants. We exploit this observation to extract the skinning component of clothing animation. To this end, we employ a simple *vertex-to-vertex deformation transfer* technique; assuming that x_j (a clothing mesh's vertex) corresponds to v_k (a character mesh's vertex) in terms of proximity (l_2 distance through a kd-tree) in their rest pose, we simply transfer v_k's bone transformations (T_b) with bone-vertex influence weights (w_{kb}) to x_j and rewrite the typical *linear blend skinning* (LBS) as $\dot{x}^i_j = (\sum_{b \in \mathcal{B}} w_{kb} T^i_b) x_j$. This approach enables us not only to easily obtain the compact representation of clothing animation by reusing the character skinning formula, but also to quickly approximate the general look of a clothing mesh in a kinematic fashion. However, the resulting deformation may suffer from a lack of fine-scale wrinkles; the fine details are diminished and smoothed out (Fig. 1(b)).

Non-skinning component with Wrinkle Flow To enrich a clothing mesh with high quality wrinkles, we handle wrinkles at the individual vertex level instead of per bone. Thus, we compute per-vertex residual vector $a^i_j = x^i_j - \tilde{x}^i_j$ and evaluate the matrix of wrinkle flow $\mathbf{A} = [a^1, a^2, \dots, a^m]$, where $a^i \in \mathbb{R}^{3n}$ (thus, $\mathbf{A} \in \mathbb{R}^{3n \times m}$). However, the wrinkle flow itself is not the ideal representation in terms of compactness. Two observations lead us to a subspace method for reducing the data size of wrinkle flow: 1) the overall appearance of the clothing mesh has already been established at the skinning stage, and 2) the residual vector field mostly handles the remaining subtle component. Thus, we do not need to retain the full data set to accurately approximate the original wrinkle flow. In this case, it is good to use principal component analysis (PCA) to reduce the data while preserving important features. In $\mathbf{A} = \mathbf{U\Sigma V}$ with SVD, by taking k principal components with the largest variance values in $\mathbf{\Sigma}$, we can obtain the projection matrix $\mathbf{U}_r = [u^1, u^2, \dots, u^k]$, where $u^i \in \mathbb{R}^{3n}$ (thus, $\mathbf{U}_r \in \mathbb{R}^{3n \times k}$), and transform the original wrinkle flow into the reduced subspace with a projection of $z^i = \mathbf{U}^T_r a^i$, where $z^i \in \mathbb{R}^k$. As a result, we obtain the matrix $\mathbf{Z} = [z^1, z^2, \dots, z^m]$, where $\mathbf{Z} \in \mathbb{R}^{k \times m}$. This method enables us to significantly reduce the data size of wrinkle flow up to $O(3n \times k + k \times m)$ from $O(3n \times m)$ because k is far smaller than m in most cases. In contrast to the previous work [Kry et al. 2002], our wrinkle flow involves the effects of wrinkle formation due to not only pose, but also dynamics variation.

Reconstruction Our animation pipeline is straightforward. We start with a clothing mesh in a rest pose. Next, we perform LBS. Then, we reconstruct wrinkle flow. Finally, we combine them together to complete the clothing animation: $\tilde{x}^i_j = \dot{x}^i_j + (\mathbf{U}_r z^i)_j = (\sum_{b \in \mathcal{B}} w_{kb} T^i_b) x_j + (\mathbf{U}_r z^i)_j$.

Conclusion We believe that our system is an advance for clothing animation, especially in CG applications such as videogames or VR simulations, where virtual characters with various costumes are commonplace and their combined animations usually predefined.

References

KRY, P. G., JAMES, D. L., AND PAI, D. K. 2002. Eigenskin: real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, SCA '02, 153–159.

*e-mail:{bukim, farbizf}@i2r.a-star.edu.sg

[1] http://www.marvelousdesigner.com

[2] http://mocap.cs.cmu.edu

SIGGRAPH 2013, July 21 – 25, 2013, Anaheim, California.

ACM 978-1-4503-2261-4/13/07