CageR: From 3D Performance Capture to Cage-based Representation

Jean-Marc Thiery   Julien Tierny   Tamy Boubekeur
Telecom ParisTech – CNRS *

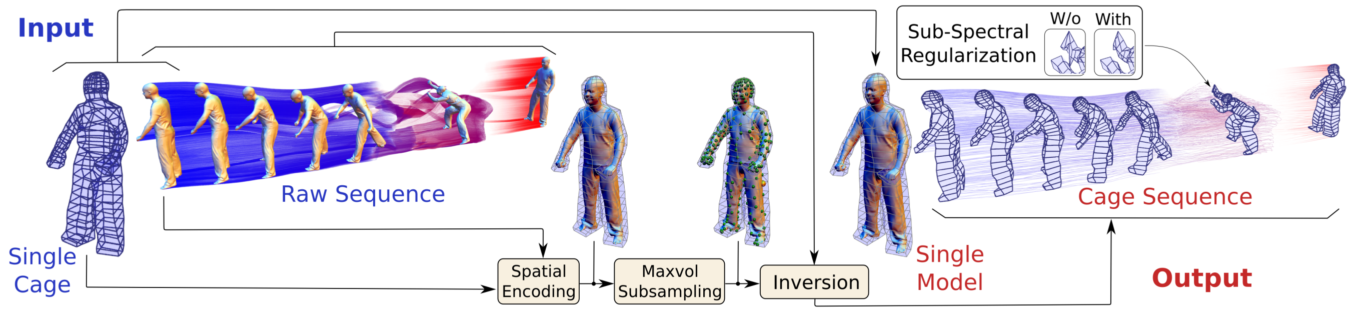

Figure 1: CAGER Processing Pipeline. *From left to right: given a raw 3D+time sequence and an initial cage, we first extract an optimal subset of positional constraints for the cage coordinate inversion. Then, the cage coordinates are inverted for each frame of the input sequence. A selective enforcement of regularization terms is defined to affect the cage vertices where the inversion is the most unstable. The resulting smoothly varying cage sequence faithfully reconstructs the input sequence when applied to a single mesh frame.*

Keywords: cage coordinates, spectral, maxvol, GPU, linear algebra, reverse engineering

1 Introduction

Modern performance capture systems [de Aguiar et al. 2008] provide high resolution 3D mesh sequences which are becoming critical components for today's special effects. Unfortunately, such raw sequences have a large memory footprint and are difficult to edit. We propose CAGER, a framework based on spatial deformation with cages to construct automatically a compact and editable high level representation of these raw sequences, resulting in high compression factors and allowing easier post processing. In particular, we formulate an automatic cage fitting algorithm embedding a new relaxation strategy based on Maximum Volume and a new regularization method based on sub-spectral analysis. As a result, we use the CAGER representation in various applications, including compression, motion transfer and shape space modeling.

2 Reverse Engineering

Given a mesh $\mathcal{M}$ and a closed triangle cage mesh $\mathcal{C}$, cage coordinate techniques, e.g. Mean Value Coordinates [Ju et al. 2005], allow to encode each vertex position p_i of $\mathcal{M}$ w.r.t. vertex positions c_j of $\mathcal{C}$ by: $p_i = \sum_j \phi_j(i) \cdot c_j$, or $\mathcal{M} = \Phi \cdot \mathcal{C}$. Given a set of poses $\mathcal{M}^k$ of the model (i.e., typical performance capture output), we want to generate a set of cages $\mathcal{C}^k$ such that $\Phi \cdot \mathcal{C}^k \simeq \mathcal{M}^k$. The L^2-projection of $\mathcal{M}_k$ onto the space of admissible deformations is $\overline{\mathcal{M}_k} = \Phi \cdot \Phi^\dagger \mathcal{M}_k$, which involves the pseudo-inverse $\Phi^\dagger$ of Φ. Unfortunately, as nowadays cage coordinate systems are *unstable to inversion* (large condition number of the system, from 10^3 to 10^{10} in the results we present), the resulting cage has very large instabilities. This set of cages is not suited for compression or any other application, as the scheme becomes sensitive to numerical errors.

To overcome this issue, we first propose to relax the system by taking only the minimum number of positions as constraints, i.e. the dimension of the problem. The selection of the so-called *handles* (green spheres in Fig. 2) is performed by looking for the *maximum volume square submatrix* $\Phi_\square$ of Φ (known as *maxvol* problem). We use the *maxvol* approximation model by Goreinov et al. [2010], for which we propose a GPU implementation. The inversion is first performed by computing the pseudo-inverse of $\Phi_\square$. Then, we add a *geometrical regularity term* on the cage geometry (e.g. minimization of the cage Laplacian) on a sub-part of the spectrum of $\Phi_\square$. This strategy automatically focuses the regularization on the cage vertices for which the inversion is the most unstable, leaving the other vertices unaffected. Then, the output cages evolve smoothly along the sequence and their initial shape features are preserved. Interestingly, our relaxation and our regularization strategies are highly compatible, as the use of *MaxVol* lowers cage instabilities and improves the spectral properties of the system to invert.

3 Results & Applications.

Some examples of CAGER reconstructions are presented in Fig. 2. Many others are provided as additional material and in the accompanying video. Before applying any subsequent data compression scheme (e.g. wavelets), our representation already offers a high compression ratio (up to 97.76%), with high numerical stability. We implemented the iterative *maxvol* approximation algorithm on GPU with CUDA, to accelerate the pre-process. Beyond compression, CAGER representations allow to transfer shape motion from captured model to synthetic ones (Fig. 2, bottom-left) and speed-up by several orders of magnitude complex geometry interpolation techniques (Fig. 2, bottom-right).

References

DE AGUIAR, E., STOLL, C., THEOBALT, C., AHMED, N., AND SEIDEL, H. 2008. Performance capture from sparse multi-view video. *SIGGRAPH 27*.

GOREINOV, S., OAELEDETS, L., SAVOSTYANOV, D., TYRTYSHNIKOV, E., AND ZAMARASHKIN, N. 2010. How to find a good submatrix. *Matrix Methods: Theory, Algorithms and Applications*, 247.

JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. *SIGGRAPH 24*, 3, 561–566.

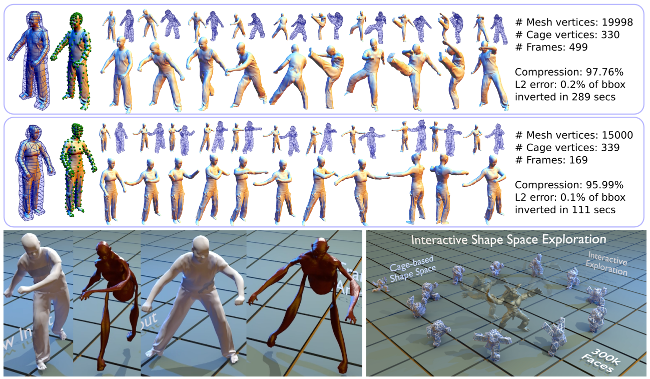

Figure 2: *Top: Results of our reverse engineering method on two performance capture data sets. Bottom-left: motion transfer. Bottom-right: interactive shape space modeling.*

[*] Email: {thiery,tierny,boubekeur}@telecom-paristech.fr

SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008