CAGER:
Cage-based Reverse Engineering of Animated 3D Shapes

Jean-Marc Thiery, Julien Tierny and Tamy Boubekeur

Telecom ParisTech - CNRS

Abstract

We present a novel framework for converting animated 3D shape sequences into compact and stable cage-based representations. Given a raw animated sequence with one-to-one point correspondences together with an initial cage embedding, our algorithm automatically generates smoothly varying cage embeddings which faithfully reconstruct the enclosed object deformation. Our technique is fast, automatic, oblivious to the cage coordinate system, provides controllable error and exploits a GPU implementation.
*At the core of our method, we introduce a new algebraic algorithm based on Maximum Volume Sub-matrices (*maxvol*) to speed up and stabilize the deformation inversion. We also present a new spectral regularization algorithm that can apply arbitrary regularization terms on selected sub-parts of the inversion spectrum. This step allows to enforce a highly localized cage regularization, guaranteeing its smooth variation along the sequence. We demonstrate the speed, accuracy and robustness of our framework on various synthetic and acquired data sets. The benefits of our approach are illustrated in applications such as animation compression and post-editing.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

Recent advances in 3D performance capture raised a large number of new problems. Among them, high level control of raw animated sequences is an important component of any editing or processing framework. While former, synthetic animated 3D sequences usually came with an underlying control structure (e.g., skeleton, cage) that tailored the animation at high level, 3D+time acquisition systems only provide raw point sets or mesh sequences, possibly consistent over time [dAST*08, PSDB*10], but without any high level control structure. Such a structure is key for compression, motion editing and other processing tasks.

High level motion control structures can be roughly classified in three categories. First, *skeletons* capture nicely locally rigid motions and are extensively used in synthetic human and animal body animation. Second, *surface handles* can be defined using a shape decomposition and help establishing a consistent segmentation of the model, allowing to morph independently each component w.r.t. rigging controllers. Third, *cages* are low resolution meshes which transfer their deformation to a high resolution model embedded in their encompassed space by the mean of a specific coordinate system. Recent advances in cage coordinate systems [JSW05, JMD*07, LLCO08] now offer a very flexible framework to smoothly and efficiently edit the shape of models with arbitrary topology, without even being constrained to have a manifold structure, using a simple cage. These three categories of structures have all their own strength and weakness, and rather complete each other than compete in a modelling and animation package.

To exploit their modelling power, with performance captured data for instance, a reverse engineering process is required to construct them automatically from a raw animated sequence. Most reverse engineering methods are built on the same observation: a large part of the raw sequence motion can be captured at a coarser level and small local motions can be ignored in a number of application scenarii. Thus, the reverse engineering process consists in replacing a sequence of (high resolution) raw models by a single static one together with a sequence of (coarser) control structures. Then, the high resolution sequence can be reconstructed by applying the animated control structure sequence to the static model. This yields immediate benefits for compression and high level editing, but also for processing and analysis, as the coarse nature of the control structure makes it practical for a number of computationally expensive techniques.

Such a reverse engineering framework has been recently proposed for skeletons [dATTHP08], where the resulting structure can be used for both processing and shape/motion modelling.

The case of cages however has not been entirely tackled, and existing methods fail to provide high quality animated cage sequences which are both able to reconstruct faithfully the input sequence and also well-structured enough to be exposed to the user for interactive post-editing.

In this paper, we address this problem and present a new powerful framework for the compact and stable encoding of deforming 3D objects with cage-based representations. Given an animated sequence (with one-to-one vertex correspondences) along with an initial cage embedding, our technique generates smoothly varying cage embeddings which reconstruct the enclosed object deformation with control-

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

lable error. Our framework is constructed around an inversion process and exhibits the following features:

- **Generality:** our approach supports synthetic and captured sequences; it is oblivious to the cage coordinate system; it can implicitly deal with realistic (e.g., as-rigid-as-possible) as well as expressive (e.g., cartoon stretching) deformations. Thus, the resulting compact cage-based representation can act as an intermediate, low-memory footprint substitute to speed up time-varying geometry processing, while delegating the tricky task of detail preservation to the underlying cage coordinate model.
- **Speed:** high resolution animated sequences are usually heavy data sets; we propose an adaptive algorithm which reduces the number of constraints taken into account, and which maps naturally to GPUs.
- **Accuracy:** we measure the reconstruction error of our cage-based representation and show that it remains low and controllable, which allows its usage for compression.
- **Usability:** when exposed to the user, each reconstructed cage should be easy to edit; we present an algorithm for localized cage regularization, yielding very smooth variations of the reconstructed cage over the sequence, while maintaining high accuracy in model reconstruction.

Technical Contributions: In order to fulfill these criteria, we propose the following contributions:

- A fast and automatic framework for the extraction of *reusable* deformation cages from arbitrary sequences of deforming 3D objects (Sec. 3).
- A purely algebraic algorithm for the extraction of optimal geometry-sensitive handles for cage deformation based on *maximum-volume sub-matrices* [GOS*10], allowing for stable and fast cage coordinate inversions (Sec. 4).
- A novel spectral regularization algorithm, enabling a localized enforcement of regularization terms, only for the cage vertices where the numerical solution of the inversion is found to be the most unstable (Sec. 5).

We evaluate our framework on various synthetic and acquired data sets and demonstrate its usefulness for different applications (Sec. 6):

- Compression: Since cages are meant to have much less vertices and faces that the enclosed model, they can be used for sequence compression. In that case, decompression consists in, given the enclosed model at the first frame only, reconstructing its poses along the sequence thanks to the cages generated by our algorithm.
- Animation Transfer: Since they are stable and smooth, the cages generated by our algorithm can be used to enclose another object (in contrast to traditional inversion strategies), thus transferring the animation to the other object.
- Speeding up geometry processing tasks: Sharp features introduced by the artist in the "rest pose cage" are well-preserved in our cages, therefore they can be used as a low-resolution representation that is suited to heavy geometry processing tasks such as mesh interpolation, as demonstrated in the accompanying video.

2. Related work

Many deformation models have been recently proposed for interactive 3D shape modeling/editing, in particular using a variational framework [LSCO*04,LSLCO05,SA07] (see the complete survey by Botsch and Sorkine [BS08]). While control skeletons [YBS07,WSLG07,JS11] are an industry standard to interact with the shape, volume deformations based on enclosing cages have become very popular due to their simplicity, flexibility and speed.

In particular, they are oblivious to the representation of the enclosed object (polygonal soup, point cloud, volumetric data etc.) and allow for efficient pre-computations, along with simple on-line computations based on linear combinations. This makes them a perfect support for deformation sequence encoding.

Cage-based Deformation: Originally investigated in 2D for scalar value interpolation on non-convex polygons [Flo03], the mean value interpolant has later been extended to closed triangle meshes embedded in 3D [JSW05], allowing for smooth volumetric deformations driven by a control cage, as well as non linear deformation speedup [HSL*06].

Since Mean Value Coordinates (MVC) involve computations based on Euclidean distances, unpleasing artifacts can occur in regions of space nearby highly concave portions of the initial cage. Harmonic coordinates (HC) [JMD*07] overcome this issue by solving Laplace equations in the cage interior. However, unlike MVC, harmonic coordinates do not have a closed form solution and can only be approximated with a numerical solver. In contrast, Green Coordinates (GC) [LLCO08, BCWG09b] admit a closed form solution (allowing efficient computations) and induce quasi-conformal (detail preserving) space deformations. Despite these advances in cage-based volume deformations, only little work payed attention to the stable computation of the inverse of the cage coordinates, a necessary step for the cage-based encoding of animated sequences.

Animation inverse kinematics: Several complete systems have been presented for the purpose of animation post-editing through the analysis of time-varying 3D shapes [SZGP05, KG06, XZY*07, KG08]. However, their internal animation representation is not compliant with industry standards (e.g., skeletons or cages), which reduces their applicative impact, as discussed in [dATTHP08].

More specialized techniques specifically address the extraction of intermediate, manipulable shape representations for the purpose of easy post-editing, such as reduced deformation models [DSP06, LWC06] or control skeletons [SC07]. However, they do not encode the deformation itself, which restricts their application to pose post-editing only. In contrast, de Aguiar et al. [dATTHP08] introduced the first framework able to fully convert the animation into a compact and editable representation. Their algorithm reconstructs a plausible skeleton corresponding to the input animation as well as the skeleton's deformation parameters enabling a

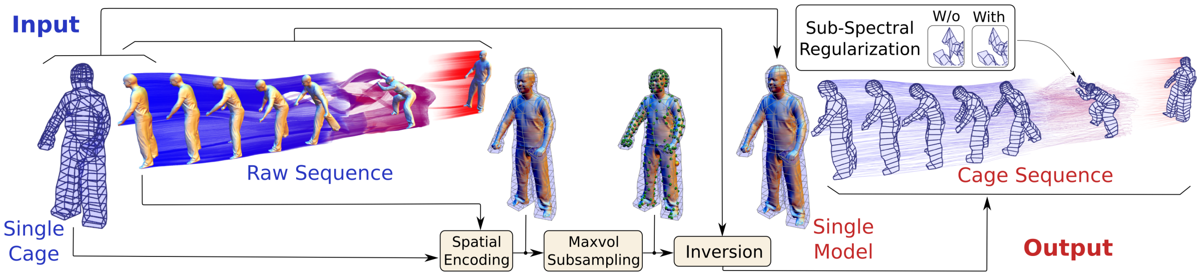


Figure 1: ***Processing Pipeline.*** *From left to right: Given a raw 3D+t sequence and an initial cage, we first extract an optimal subset of positional constraints for the cage coordinate inversion. Then, the cage coordinates are inverted for each frame of the input sequence. A selective enforcement of arbitrary 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.*

faithful reconstruction of the input sequence, useful for compression for instance. However, such techniques are mostly restricted to rigid motion estimation, suitable for human bodies for instance, but not for sequences subject to stretching or volume variation. More importantly, such methods are exclusively surface-based and require a manifold input. Also, skeletons are mostly suited for shapes exhibiting tubular components and they are difficult to extend to arbitrary shapes. A few skinning approaches [JT05, KSO10] allow to reverse-engineer an animated mesh by considering moving 3D frames instead of the skeleton's bones. This flexibility allows to process sequences including non-trivial motion, like cloth motion for instance [KSO10]. However, the absence of a global structure on these *frames* limits the spectrum of their applications to compression.

Only a few approaches address the stable and efficient inversion of cage-based deformations. Xu et al. [XZY*07] propose to transfer an initial cage embedding from one frame of the sequence to the others. However, their algorithm does not explicitly invert the transformation, but instead tries to compensate the cage pose change through local rotation blendings, which turns to be inaccurate and unstable, especially when the cage is not close to the object. Similarly, a possible strategy could consist in transferring the motion from the mesh to the cage, using deformation transfer algorithms [SP04]. However, this strategy would induce a large reconstruction error of the model by the deformed cage since there is no guarantee in general that the cage reproduces in its interior the transformation it undergoes. For instance, an as-rigid-as-possible (ARAP) transformation of the cage does not induce an ARAP transformation of the enclosed model and reciprocally. The same remark goes for any direct spatial encoding of the cage w.r.t. the model geometry, such as maintaining the cage as an offset of the animated mesh for example. Moreover, the inversion of the cage *must* take into account the coordinate system used for reconstruction. Indeed, for a given cage, reconstructing an enclosed object with different coordinates (Mean Value Coordinates, Green Coordinates, Harmonic Coordinates, Positive Mean Value Coordinates, Maximum Entropy Coordinates, etc.) leads to different results.

Ben-Chen et al. [BCWG09b, BCWG09a] automatically compute new cage embeddings to satisfy sparse user constraints through a sequence of least squares fitting. However, their framework is using GC only and enforces pure rotations on the *medial axis* of the shape, as well as Hessian's norm minimization on the cage, and is highly restricted to as-rigid-as-possible deformations and does not specifically address general and stable cage inversion.

Savoye et al. [SF10] introduce a global regularization term aiming at preserving the differential coordinates of the cage vertices. Being not rotation invariant, such an approach leads to shrinking artifacts and fails at preserving the cage geometry, which is often carefully designed by artists and critical for editing. Finding an efficient but general regularization term, with no a priori on the input sequence, and which does not completely smooth away the cage geometry, turns out to be unexpectedly difficult. We cope with it by reshaping the linear least squares fitting pipeline with a novel, efficient inversion algorithm, coupled with a new selective regularization scheme, which only regularizes the cage vertices where the inversion is less numerically stable.

3. Overview

Input: We consider a raw animated sequence of N frames, represented by a set of points $\mathcal{P}$ with time-varying embeddings in $\mathbb{R}^3$ $(\mathcal{P}_1, \mathcal{P}_2, \dots \mathcal{P}_N)$. This position data can correspond to the embedding of a triangle mesh, a point cloud or a volumetric mesh. Additionally, we consider an input cage $\mathcal{C}_R$ enclosing an arbitrary reference frame $\mathcal{P}_R$ of the sequence.

Inversion procedure: Our processing pipeline (Fig. 1) is composed of three major stages:

1. Compute the spatial encoding Φ of $\mathcal{P}_R$ into $\mathcal{C}_R$ using a cage coordinate system (e.g., MVC, HC, GC).
2. Identify the set of handles $\mathcal{H}_R \subset \mathcal{P}_R$ which maximizes the volume of a squared version of the cage coordinate matrix. This property stabilizes and speedups the inversion while guaranteeing dense inversion spectra.
3. For each frame $\mathcal{P}_t$, the corresponding cage embedding is computed by inverting the cage coordinates, using $\mathcal{H}_t$ (embedding of $\mathcal{H}_R$ at frame t) as positional constraints.

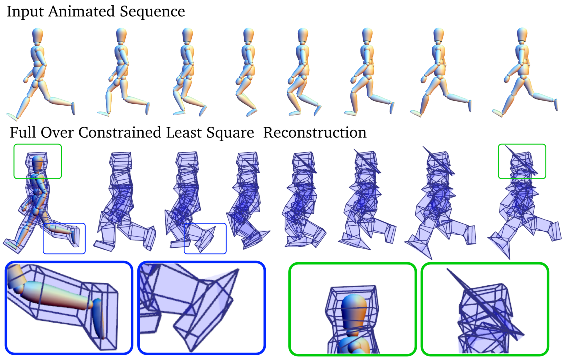


Figure 2: *Cage coordinate (MVC) inversion using a traditional over-constrained least-squares fitting results in unstable cage inversions (middle row). The output cages exhibit spikes and other artifacts (bottom row), which makes them unusable for manual post-editing.*

Additionally, our framework can selectively apply arbitrary regularization terms on the highest frequencies of the spectrum of the inversion (the last singular vectors of the coordinate matrix singular value decomposition).

Output: The resulting set of smoothly varying cages compactly and faithfully reconstructs the input sequence.

4. MaxVol based Cage Inversion

In the following, we present our algorithm for the stable inversion of the cage coordinates. We briefly discuss why a simple strategy based on over-constrained least-squares fitting can be unsatisfactory. Then we present our algebraic algorithm based on optimal handle identification.

4.1. Problem statement

Given an enclosed 3D model represented by its reference position data $\mathcal{P}_R$ and a closed triangle cage mesh $\mathcal{C}_R$, cage coordinates (such as MVC [JSW05], HC [JMD*07] or GC [LLCO08]) allow to encode each point position $p_i \in \mathbb{R}^3$ of $\mathcal{P}_R$ w.r.t. to the cage vertex positions $c_j \in \mathbb{R}^3$ (and triangle normals n_j for GC) of $\mathcal{C}_R$ by: $p_i = \sum_j \phi_j(i) \cdot c_j$, or $\mathcal{P}_R = \Phi \cdot \mathcal{C}_R$, where $\mathcal{P}_R$ is represented as an $(n \times 3)$-matrix, Φ is a rectangular $(n \times m)$-matrix and $\mathcal{C}_R$ is a $(m \times 3)$-matrix.

Given a set of poses $\mathcal{P}_t$ of the model, we aim at computing a set of corresponding cages $\mathcal{C}_t$, such that $\Phi \cdot \mathcal{C}_t \simeq \mathcal{P}_t$. The L_2-projection of $\mathcal{P}_t$ onto the space of admissible deformations is $\overline{\mathcal{P}_t} = \Phi \cdot \Phi^\dagger \mathcal{P}_t$, which involves the pseudo-inverse $\Phi^\dagger$ of Φ. As the cage coordinate matrix Φ is large and dense, the computation of its pseudo-inverse $\Phi^\dagger$ (an $(m \times n)$-matrix) through singular value decomposition (SVD) is expensive. A faster approach consists in solving the equivalent system, where $(\Phi^T \cdot \Phi)^\dagger$ is an $(m \times m)$-matrix:

$$\mathcal{C}_t = (\Phi^T \cdot \Phi)^\dagger \Phi^T \mathcal{P}_t \tag{1}$$

This yields an over-constrained linear system, where the number of unknowns m (the number of vertices in C_t for MVC, plus the triangle normals for GC) is meant, by definition of control cages, to be significantly smaller than the number of constraints n (the number of point samples in the model). Let $U \Sigma V^T$ be the SVD of $(\Phi^T \cdot \Phi)$. The solution is given by:

$$c_{j_t} = \sum_k V_{jk} \cdot \left((U^k)^T \cdot \Phi^T \cdot \mathcal{P}_t \right) \cdot \frac{1}{s_k} \tag{2}$$

with s_k the k^{th} singular value.

However, a slight perturbation $\delta_{\mathcal{P}_t}$ on the constraints $\mathcal{P}_t$ may be drastically amplified through the SVD when projecting onto the singular vectors associated with low singular values (Eq. 2). We illustrate this phenomenon in Fig. 2: over-constrained least-squares fitting yields instabilities resulting in important spikes on the output cages, which makes them useless application-wise. To overcome this issue, we first propose a strategy to optimally relax the system, which has the beneficial side-effect of decreasing the condition number of the coordinate matrix, hence stabilizing its inversion.

4.2. MaxVol relaxation

Reducing the number of constraints of an over-determined linear system reduces the chances of taking into account multiple conflicting constraints. Traditionally, an arbitrary subset of the constraints is considered, whose size is progressively reduced until a satisfactory trade-off between stability and precision is obtained. Ultimately, one could narrow the size of the constraint set down to that of the unknowns. However, as shown in Fig. 3, this selection process must be carefully carried out to maintain a decent solution precision.

Minimum condition number sub-matrix To optimally reduce the number of constraints, while guaranteeing a stable inversion process, one needs to seek the optimal square $(m \times m)$ sub-matrix $\Phi_\square$ of Φ, with minimum condition number. This linear algebra problem is well known to be NP-hard. As discussed by several authors [CMI07, ĒMI09, GOS*10], computing ε-approximations is NP-hard as well.

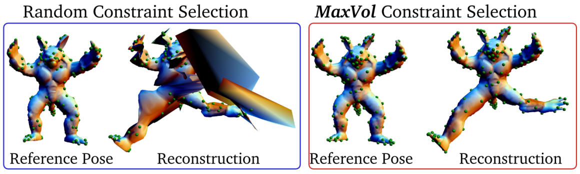


Figure 3: *Minimal selection of position constraints (as many as cage vertices, MVC). Selecting constraints randomly on the reference frame $\mathcal{P}_R$ (left) generates a cage for the other frames which performs poor reconstruction (middle left). In contrast, our relaxation strategy (with the same amount of constraints) generates a smooth reconstruction (right).*

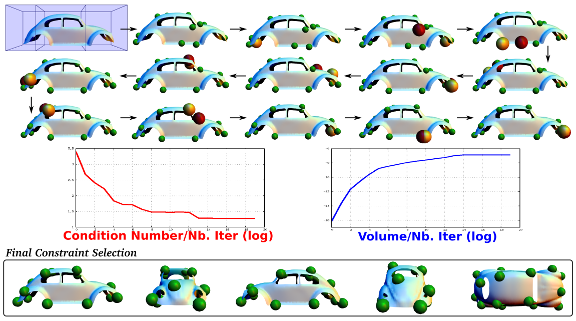


Figure 4: *Each iteration of the maxvol algorithm (top) corresponds to deselecting a constraint (red), and selecting another (yellow). The condition number and the volume of $\Phi_\square$ are respectively shown in red and blue (log scale).*

Maximum volume sub-matrix A weaker indication of the *invertibility* of a matrix is given by the value of its determinant, which ought to be high for the matrix to admit stable inversion. Finding a square sub-matrix that has maximum volume (absolute value of the determinant) has been intensively researched by the linear algebra community and an efficient iterative algorithm has been proposed recently [GOS*10]. Although it is not guaranteed to identify the optimal sub-matrix maximizing its volume, its practical performances demonstrate significant decrease of the condition number, which in the worst case falls back to that of the over-determined system. In the following, we sketch the main steps of this algorithm and detail its integration into our framework.

MaxVol algorithm Let A be an $(n \times m)$-matrix $(m < n)$. The goal of the algorithm is to identify the m rows of A such that the resulting square $(m \times m)$ sub-matrix has maximum volume. The algorithm *MaxVol* is based on simple observations, further discussed in the original paper [GOS*10].

Let $A_\square$ be the top square sub-matrix of A, composed of the first m rows of A, and $B = A \cdot A_\square^{-1}$. Swapping the rows $i > m$ and j in A for which B_{ij} has maximum absolute value increases the volume of the top square sub-matrix of A, if $|B_{ij}| > 1$. Let e_j be the column vector with value 1 at j and 0 elsewhere. Then, both swapping the rows i and j in A and updating B can be performed in a single update of the form:

$$B := B - (B_{:j} - e_j + e_i) \cdot (B_{i:} - e_j^T)/B_{ij} \tag{3}$$

where $B_{:j}$ depicts the entire j^{th} column of B and $B_{i:}$ its entire i^{th} row.

This yields a practical iterative algorithm (Fig. 5):

- **Initialization:** Order the rows of A, such that its top square sub-matrix $A_\square$ is invertible. Compute $B = A \cdot A_\square^{-1}$.
- **Iteration:** Find the maximum absolute value element B_{ij} with $i > m$. Swap rows i and j in the current solution. Update B with Eq. 3.

The iterations can be stopped on demand or when the maximum B_{ij} gets smaller than one. In terms of complexity, the initialization stage takes $O(m^2 \times n)$ operations and each iteration takes $O(m \times n)$ steps.

4.3. Cage inversion based on maxvol relaxation

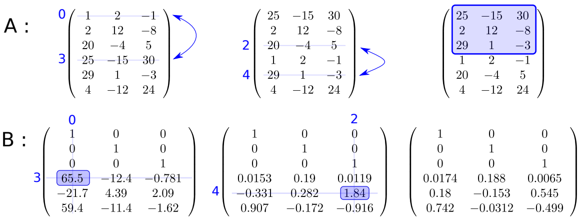
Applying the maxvol algorithm to our setting is straightforward, the input being Φ and the output being $\Phi_\square$. In practice, we observed that about $2 \times m$ iterations are required until convergence. Fig. 4 illustrates a few iterations of the algorithm. Both the volume and the condition number of $\Phi_\square$ respectively increases/decreases at each iteration (results are shown in log scale), assessing the *invertibility* quality of $\Phi_\square$.

Cage coordinate inversion For each frame $\mathcal{P}_t$, the corresponding cage embedding $\mathcal{C}_t$ is given by (where $\mathcal{H}_t$ is the sub-matrix of the top m rows of $\mathcal{P}_t$):

$$\mathcal{C}_t := \Phi_\square^\dagger \mathcal{H}_t \tag{4}$$

The pseudo-inverse $\Phi_\square^\dagger$ is computed through SVD, as discussed in the case of $(\Phi^T \cdot \Phi)^\dagger$ in Eq. 2 ($\Phi^T \cdot \mathcal{P}_t$ needs to be exchanged with $\mathcal{H}_t$). Notice that for each frame, the rows of $\mathcal{C}_t$ and $\mathcal{P}_t$ also need to be swapped according to the swaps performed in Φ by maxvol. Fig. 6 provides a comparison of the repartition of the singular values yielding from the SVD, between our technique ($\Phi_\square$) and the classical over-determined approaches (Φ and $\Phi^T \cdot \Phi$). Notice that $\Phi_\square$ exhibits a more *convex* spectrum. Also, as discussed with Eq. 2, important instability occurs in the inversion if the last singular values of the SVD are low. With our relaxation scheme, the last singular values of $\Phi_\square$ are much higher than those of the traditional over-determined approximations (Fig. 6, rightmost plots), which yields much more stable inversions.

Finally, once $\Phi_\square$ is computed, the inversion of the cage embeddings for the whole sequence is faster with our approach than with the traditional over-determined least-square strategy. In both cases, the SVD of an $(m \times m)$ matrix needs to be computed in a pre-process. For each frame, with our strategy, the solution is obtained by the multiplication of an $(m \times m)$ by an $(m \times 3)$-matrix (Eq. 4). For the traditional least-squares approach however, an extra multiplication between an $(m \times n)$ and an $(n \times 3)$-matrix is required (Eq. 1).



$$A: \quad \begin{array}{r} 0 \\ \\ \\ 3 \\ \\ \\ \end{array}\begin{pmatrix} 1 & 2 & -1 \\ 2 & 12 & -8 \\ 20 & -4 & 5 \\ 25 & -15 & 30 \\ 29 & 1 & -3 \\ 4 & -12 & 24 \end{pmatrix} \quad \begin{array}{r} \\ \\ 2 \\ \\ 4 \\ \\ \end{array}\begin{pmatrix} 25 & -15 & 30 \\ 2 & 12 & -8 \\ 20 & -4 & 5 \\ 1 & 2 & -1 \\ 29 & 1 & -3 \\ 4 & -12 & 24 \end{pmatrix} \quad \begin{pmatrix} 25 & -15 & 30 \\ 2 & 12 & -8 \\ 29 & 1 & -3 \\ 1 & 2 & -1 \\ 20 & -4 & 5 \\ 4 & -12 & 24 \end{pmatrix}$$

$$B: \quad \begin{array}{r} \\ \\ \\ \\ 3 \\ \\ \\ \end{array}\begin{array}{ccc} 0 & & \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \boxed{65.5} & -12.4 & -0.781 \\ -21.7 & 4.39 & 2.09 \\ 59.4 & -11.4 & -1.62 \end{array} \quad \begin{array}{r} \\ \\ \\ \\ \\ 4 \\ \\ \end{array}\begin{array}{ccc} & & 2 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.0153 & 0.19 & 0.0119 \\ -0.331 & 0.282 & \boxed{1.84} \\ 0.907 & -0.172 & -0.916 \end{array} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.0174 & 0.188 & 0.0065 \\ 0.18 & -0.153 & 0.545 \\ 0.742 & -0.0312 & -0.499 \end{pmatrix}$$

Figure 5: *Iterations of the* maxvol *algorithm on a toy example. At each iteration, the location of the element B_{ij} with maximum absolute value indicates which rows to flip in A to improve the volume of $A_\square$ (in blue on the right).*

Geometric interpretation Although our strategy is purely algebraic, it yields interesting geometrical insights. When reflected on the matrix $\mathcal{P}_R$, the row swaps of Φ are equivalent to deselecting a constraint and picking another one instead (Fig. 4). Near convergence, the constraints to flip get closer and closer. At the end of the process, the set of corresponding points of $\mathcal{P}_R$ (in green in Fig. 4) can be interpreted as a minimally optimal set (noted $\mathcal{H}_R \subset \mathcal{P}_R$) of handles for deformation, given the model $\mathcal{P}_R$, the cage $\mathcal{C}_R$ and the employed cage coordinate model. As shown in Fig. 7, the local density of $\mathcal{C}_R$ is captured by $\mathcal{H}_R$. Also, configurations implying high curvature, sharp edges, or boundaries (Fig. 4) are captured by $\mathcal{H}_R$, while the symmetries of the cage are also represented by $\mathcal{H}_R$. In other words, the constraints $\mathcal{H}_R$ are well dispatched in the space of cage coordinates.

Fig. 8 shows the cage reconstruction with MVC of two poses with our inversion strategy, compared to the classical over-determined least-squares. Since our strategy relaxes the set of constraints, the error in the *model* reconstruction is necessarily slightly higher. However, as shown on the right, our approach removes many of the *cage* instabilities occurring with the over-determined approach.

Coordinate system analysis Fig. 9 presents the spectral behavior of the different cage coordinate systems used in the paper, namely MVC, HC and GC. MVC and HC exhibit similar spectral behavior: in comparison to the over-determined full least-squares system (Fig. 9, bottom), MaxVol exhibits much higher last singular values, which stabilizes the inversion. In the case of GC, MaxVol increases the singular values all over the spectrum. In all the cases, MaxVol decreases significantly the condition number of the system.

Fig. 10 shows the cage reconstruction with our inversion strategy (MaxVol or not), with different coordinate systems (MVC, HC, GC). Since they do not admit a close form expression, HC need to be *approximated* with a solver, which necessarily leads to *residuals* (as disccused in [JMD*07]), whose importance is implementation-dependant. This imprecision can be observed when inverting the reference pose $\mathcal{P}_R$ (Fig. 10, right): the inversion does not result in the identity (spike on the base of the tail). The GC are the most instable to invert. The space of transformations they allow is too large to allow accurate inversions if the face normals are not constrained to be orthogonal to the cage faces. Also, the inversion uniformly processes all the unknowns of the system whereas for GC, these mix much different entities (vertex positions *and* face normals).

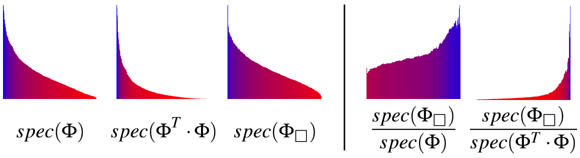


Figure 6: *Comparison of the coordinate matrix spectra for the sequence of Fig. 1 with MVC (normalized plots). Our approach ($\Phi_\square$) exhibits a more* convex *spectrum, with higher last singular values (right), yielding more stable inversions.*

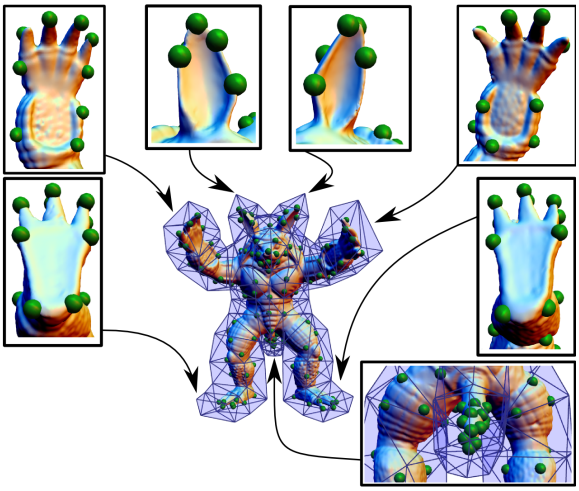


Figure 7: *Geometry-sensitive deformation handles extracted by our **algebraic-only** algorithm. Sharp edges (ears) and high curvatures (finger tips) are implicitly detected as constraints. The sampling density (tail) and the symmetry of the cage are also captured by the constraints.*

As shown in Fig. 8, our relaxation strategy significantly reduces the condition number of the system, resulting in less instabilities on the inverted cages. Still, a few of them may remain (Fig. 8, right). To overcome this issue, we introduce a novel regularization strategy, that selectively focuses on the unknowns for which the most instabilities occur.

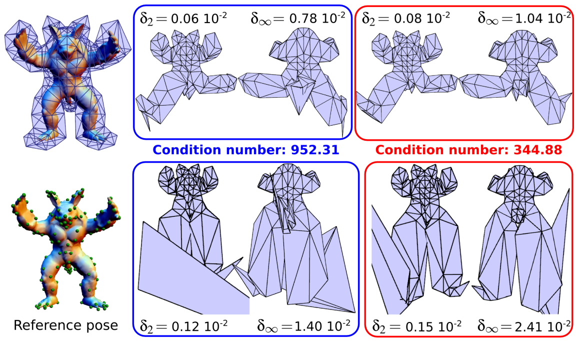

$\delta_2 = 0.06\ 10^{-2}$
$\delta_\infty = 0.78\ 10^{-2}$
$\delta_2 = 0.08\ 10^{-2}$
$\delta_\infty = 1.04\ 10^{-2}$
Condition number: 952.31
Condition number: 344.88
Reference pose
$\delta_2 = 0.12\ 10^{-2}$
$\delta_\infty = 1.40\ 10^{-2}$
$\delta_2 = 0.15\ 10^{-2}$
$\delta_\infty = 2.41\ 10^{-2}$

Figure 8: *Comparison between the over-determined least-squares approach (blue) and our relaxation strategy (red) for deformations based on rotations (top) and stretching (bottom). The point-to-point L_2 and L_∞ distances between the reconstructed model and $\mathcal{P}_t$ are given by δ_2 and δ_∞ (fraction of the model bounding box diagonal).*

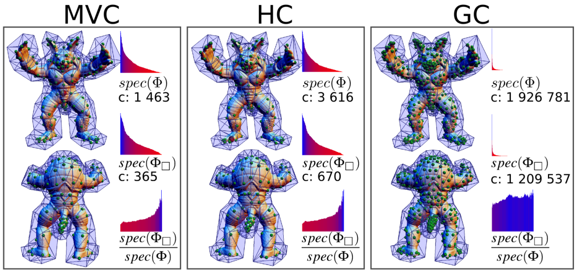


Figure 9: *MaxVol handles and coordinate matrix spectra for different coordinate systems (MVC, HC, GC). In each case, MaxVol drastically reduces the condition number of the system ($\Phi_\square$, middle) in comparison to over-determined least-squares (Φ, top). The gain (bottom row) is maximum in high frequencies, where the inversion is the most instable.*

5. Sub-spectral regularization

Traditionally, to cope with the instability of a linear system, a solution consists in introducing *globally* a *geometrical* regularization term. This enables to bias the solution towards a space of preferred solutions (e.g. detail-preserving or as-rigid-as-possible transformations). However, global regularization also comes with several drawbacks. First, such a solution is not general, since it makes assumptions on the transformations present in the input sequence. Second, even with a priori information on the model transformations, there is no direct connection to the transformations that the cage should undergo. Third, in practice, this solution tends to over-damage the data fitting, even before all of the instability is corrected.

5.1. Regularization terms

In our setting, without any a priori on the space of transformations exhibited in the input sequence, one of the only desirable properties of a regularization term is its invariance against rotation. For instance, Savoye et al. [SF10] enforce the preservation of the differential coordinates of the cage vertices. However, this is not rotation invariant and leads to significant shrinking artifacts. The preservation of the norm of the Laplacian of the cage vertices does not suffer from this drawback but involves non linear terms [ESP08].

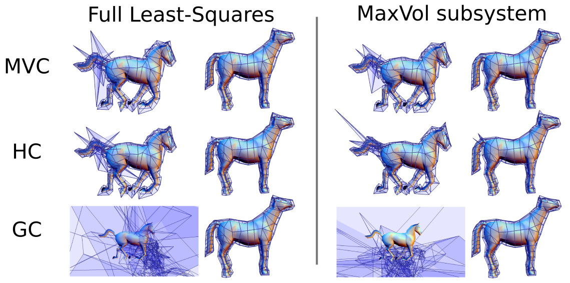


Figure 10: *Comparison between over-determined and MaxVol inversions with different coordinate systems.*

Minimizing the Laplacian of the cage vertex positions is rotation invariant but, without sophisticated reconstruction strategy [BK03], it smooths away the cage curvature. The same holds for the minimization of the deformation Hessian [BCWG09b, TTB11]. In other words, rotation invariant linear regularization terms are *destructive*, in the sense that they will smooth away the information provided in the input cage $\mathcal{C}_R$. In the following, we present a new regularization algorithm that locally allows the usage of *destructive* terms only for the unknowns where the inversion solution is unstable and not reliable.

5.2. Sub-spectral regularization algorithm

Let $U\Sigma V^T$ be the SVD of $\Phi_\square$. The vector of the cage vertex positions (and of its face normals for GC) can be expressed as a linear combination of the columns of V: $\mathcal{C}_t = \sum_k V^k \cdot q_{k_t}$, $q_{k_t} \in \mathbb{R}^3$. The solution to the system inversion is given with $q_{k_t} = (U^k)^T \cdot \mathcal{H}_t / s_k$. A common strategy for sub-spectral regularization [HSS92] consists in directly cropping the spectrum by setting the inverse of the last singular values to 0. Instead, a second common solution [HSS92] consists in parameterizing the stability of the solution by a scalar α, and using s'_k instead of s_k, where $s'_k = \frac{s_k}{s_k^2+\alpha^2}$. These two strategies do not allow for the insertion of a geometrical regularization term, which would take into account the physics of the problem. In the following, we introduce a novel and general sub-spectral algorithm, which allows to include relevant geometrical regularization terms only on the most instable portions of the spectrum of $\Phi_\square$.

As shown in Fig. 6, the left part of the spectrum of $\Phi_\square$ can be interpreted as the low frequencies of the inversion, while the right part can be interpreted as its high frequencies. Also, as discussed earlier, the *cage* reconstruction instability is more likely to occur on the high frequencies of the spectrum,

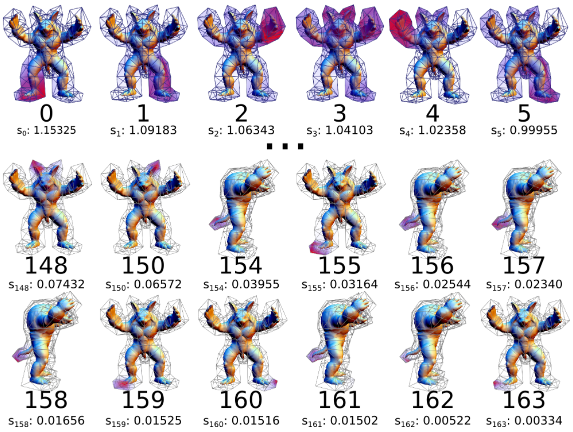


Figure 11: *Singular vectors of $\Phi_\square$ on the cage (MVC, in absolute value, gradient of pink). For the last singular values, where instabilities may occur, these vectors are localized to a small subset of cage vertices.*

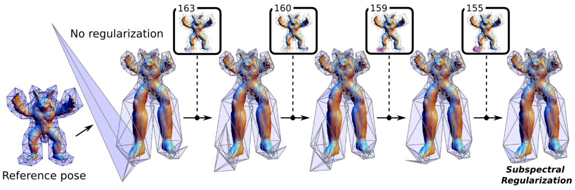


Figure 12: *Cage and model reconstruction with MVC and Laplacian minimization. Increasing the* spectrum threshold *yields localized and progressive blending of the unstable cage vertex solutions with the solutions of the regularization.*

where the singular values are low. The last singular vectors of $\Phi_\square$ correspond to its *"pseudo-kernel"*, in the sense that a perturbation in that space induces only a slight change in the *model* reconstruction, since it is amplified by the corresponding singular value. In other words, relaxing the data-fitting constraints for these vectors will have a negligible impact on the reconstruction of the model. Therefore, our technique enforces regularization terms (expressed through a matrix Δ) only on the high frequencies of the spectrum. As the last singular vectors are highly localized (see Fig. 11), this guarantees a minor *destructive* impact of the regularization term on the global aspect of the cage inversion (its low frequency component) while locally fixing instabilities.

Given a *spectrum threshold s*, the low frequency part of the spectrum is directly reconstructed from the positional constraints $\mathcal{H}_t$:

$$q_{k_t} = (U^k)^T \cdot \mathcal{H}_t / s_k, \qquad \forall k < (m-s) \tag{5}$$

The high frequency part is reconstructed with the traditional combination of positional constraints and regularization, to minimize the following energy ($\lambda \in [0,1]$):

$$E = \lambda ||\Delta \cdot \mathcal{C}_t||^2 + ||(\Phi_\square \cdot \mathcal{C}_t) - \mathcal{H}_t||^2 \tag{6}$$

which is equivalent to (with q_{k_t} fixed $\forall k < (m-s)$):

$$E = \lambda || \sum_{k<(m-s)} \Delta \cdot V^k \cdot q_{k_t} + \sum_{k \geq (m-s)} \Delta \cdot V^k \cdot q_{k_t} ||^2 + ||(\Phi_\square \cdot \mathcal{C}_t) - \mathcal{H}_t||^2 \tag{7}$$

This is equivalent to minimizing the following energy, which can be done by inverting a linear system in the least-squares sense:

$$\lambda || \sum_{k \geq (m-s)} \Delta \cdot V^k \cdot q_{k_t} + \sum_{k<(m-s)} \Delta \cdot V^k \cdot \left((U^k)^T \cdot \mathcal{H}_t / s_k \right) ||^2 + \sum_{k \geq (m-s)} s_k^2 ||q_{k_t} - \left((U^k)^T \cdot \mathcal{H}_t / s_k \right) ||^2$$

This strategy can be used with arbitrary linear regularization terms expressed through a matrix Δ. Figs. 12 and 13 show examples of cage and model reconstructions with a particularly destructive term, the minimization of the cage Laplacian. As shown in Fig. 12, our spectral strategy enables a progressively localized blending of the data-fitting constraints with the regularization term, while the rest of the cage remains unaffected. Fig. 13 provides further comparisons between a classical regularization strategy and our spectral approach on a challenging cage with sharp dihedral angles and a deformation with high distortion. The color map plots the point-to-point distance with the non-regularized cage. In other words, it depicts the local quality of the data fitting in term of *cage* reconstruction. Whereas classical regularization damages globally the reconstructed cage, our strategy automatically localizes the effect only on the unstable cage vertices, while preserving the *true* solution on stable ones, yielding much more accurate *model* reconstruction.

6. Results

In this section, we present the results of our technique on data-sets coming from spatio-temporally coherent motion captures [dAST*08] and synthetic animations (see Fig. 14). These experiments have been run under Linux on a commodity laptop with a Core 2 Duo 2.53 GHz CPU and a GeForce GTX 260 M GPU. Our programs are written in C++. The maxvol algorithm is fully implemented on the GPU with CUBLAS, while SVD computation uses GSL and OpenNL is used to solve Eq. 7. Unless explicitly mentioned, all the results have been computed with the same default setting (MVC, MaxVol selection, Laplacian minimization, $\lambda = 1$, $s = 20$). Distance measurements are expressed w.r.t. the bounding box diagonal.

6.1. Encoding quality

Fig. 14 shows the cage-based encoding of several input sequences, along with the resulting reconstruction. The corresponding timings are provided in table 1. Note, that the

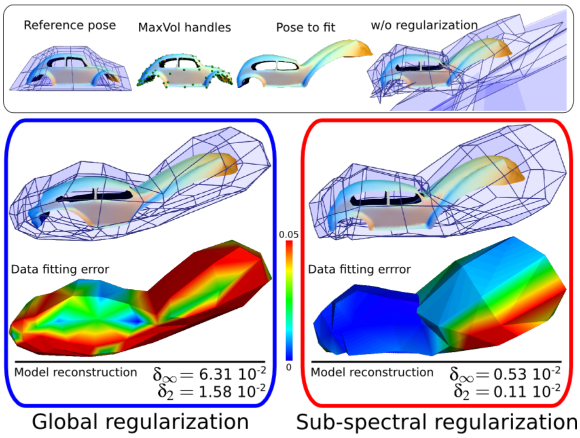


Figure 13: *Localized effect of our sub-spectral regularization ($s = 20$), in comparison to traditional global regularization, on a* destructive *regularization term: the Laplacian minimization (MVC and maxvol relaxation). In both cases, the regularization weight λ is set to the same value (1).*

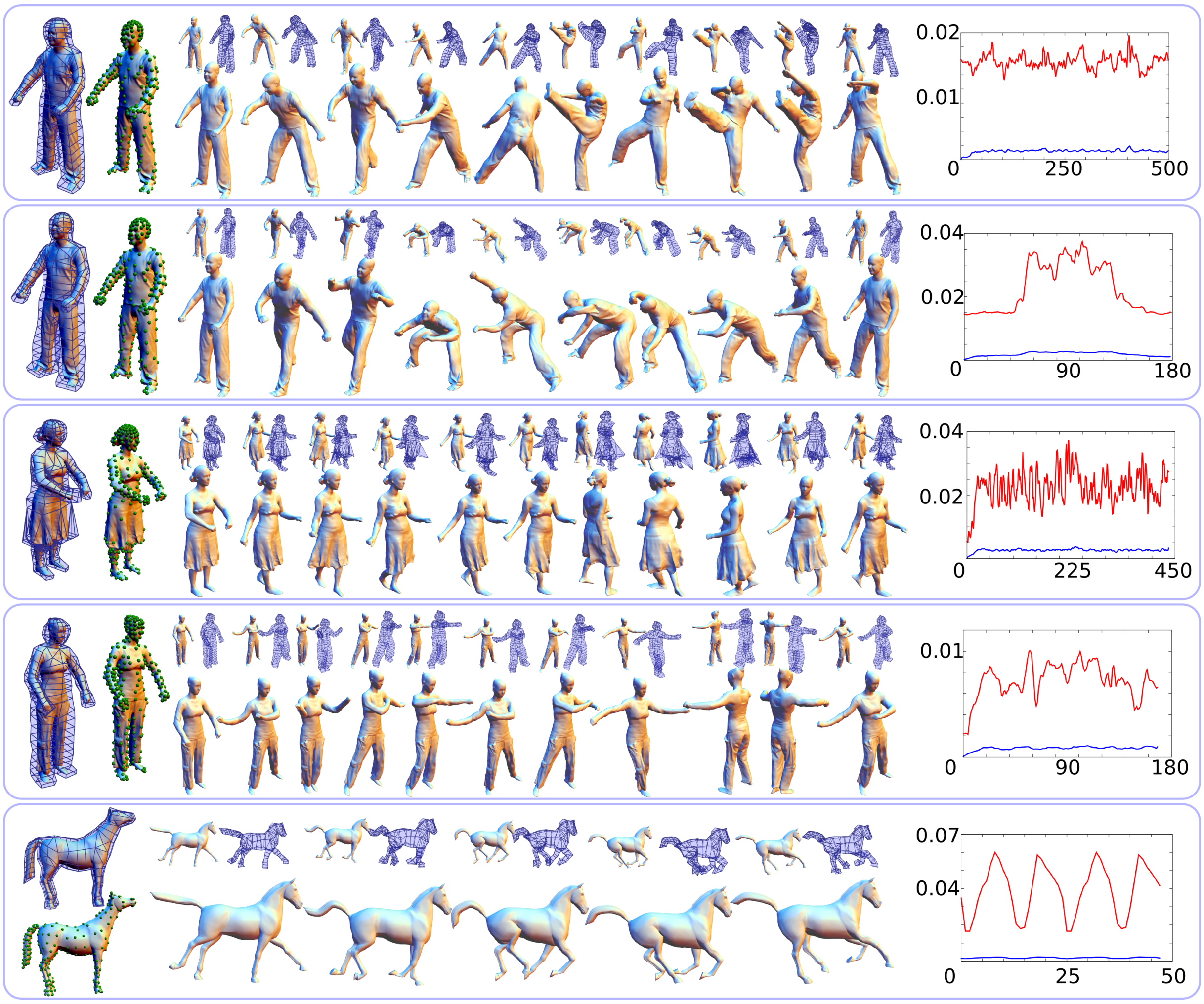


Figure 14: ***Reconstructed sequences:*** *for each sequence, $\mathcal{P}_R$, $\mathcal{C}_R$ and $\mathcal{H}_R$ are shown on the left. For each frame t, on the top, $\mathcal{P}_t$ appears on the left; the reconstructed cage is on the right, the reconstructed frame is at the bottom (larger view). The blue and red curves respectively show the evolution of the point-to-point L_2 and L_∞ model reconstruction error (x-axis: frame number).*

cage embeddings smoothly vary over time (cf. **video**), while guaranteeing low model reconstruction error. Hence, the output cages can be used for post-editing. The algorithm, without any geometrical a priori, implicitly handles acquired data-sets (top-4 rows) and synthetic data motioned through as-rigid-as-possible (bottom row) or cartoon-like stretching deformations (Fig. 12). The algorithm successfully reconstructs the sequence despite chaotic deformations (cf. the skirt motion, 3rd row). Note, that in this specific example, the spectral regularization automatically removed unstable cage spikes but preserved those which are mandatory for a correct reconstruction.

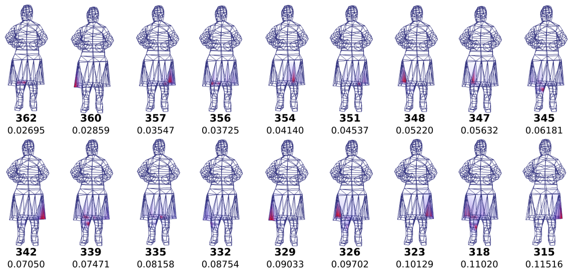


Figure 15: *Singular vectors (with their singular values) localized on the skirt of the* Skirt *model. A change in the 315^{th} singular vector induces a change in the model reconstruction, up to a factor 1/10.*

As demonstrated in Fig. 15, removing systematically all the spikes appearing on the skirt would require a regularization up to the 50^{th} last singular vectors. However, the corresponding singular value is then more than 0.1, which would induce significant model reconstruction errors. Then, the remaining spikes appearing in Fig. 14 are necessary for an ac-

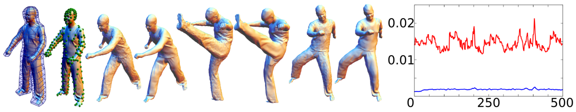


Figure 16: *Reconstruction (right) of a noisy sequence (left).*

curate model reconstruction. Moreover, these features of the cage still evolve smoothly along the sequence (cf. **video**).

6.2. Encoding robustness

Fig. 16 shows a reconstruction example with random perturbations inserted in the sequence to fit (each vertex is displaced randomly within a radius equal to 0.3×10^{-2} times the bounding box diagonal). The algorithm still generates a valid reconstruction. Note that the curves of the model reconstruction errors exhibit the same overall behavior than those of the original data-set (Fig. 14, top row) but with additional noise. In particular, the difference between the maximum L_∞ errors is 0.36×10^{-2}, which exactly reflects the introduced noise and which further assesses the robustness of the algorithm.

Fig. 17 shows a reconstruction with a reference cage computed **automatically** (602 vertices). Interestingly, the reconstruction error is lower (with an average L_∞ error of 0.73×10^{-2} against 1.39×10^{-2} in Fig. 14, top row). Indeed, realistic cages designed by artists (Fig. 14) are more challenging to fit, given their specific structure and the very small number of unknowns they provide to the inverse system.

6.3. Comparisons and limitations

The localized nature of our sub-spectral regularization allows to reconstruct a cage sequence which is more faithful to the input reference cage, avoiding the rounding artifacts stemming from *destructive* regularization terms (Fig. 13). Fig. 18 provides a comparison between our algorithm and a traditional, globally regularized, over-determined least-squares (with the exact same setting). Important artifacts occur with the traditional approach (see the inset zooms), yielding higher reconstruction error (with an average L_2 error of 0.24×10^{-2} against 0.14×10^{-2} with our approach).

As discussed earlier, not only our sub-spectral scheme enables localized regularization, but it also automatically focuses on the cage vertices where higher instabilities occur.

Fig. 19 and 20 show the relative importance of each step of our approach (*MaxVol* and *spectral regularization*) on a toy example and the *Skirt* sequence respectively. In Fig. 19, the upper part of a bumpy torus has been scaled up (left column). In contrast to the other methods, cage inversion by MaxVol relaxation and spectral regularization reconstructs a cage with a visually similar deformation. In Fig. 20, the regularization parameters have been increased (s = 50) for illustration purpose. The full least-squares approach produces the largest spikes (1^{st} row). These are stabilized thanks to the MaxVol relaxation (2^{nd} row). Spectral regularization further improves the cage stability (3^{rd} and 4^{th} row), while the most stable cages are obtained in conjunction with MaxVol relaxation (4^{th} row).

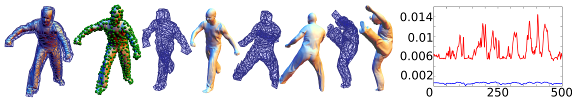


Figure 17: *Encoding with a reference cage* ***generated automatically****. The surface is first voxelized. We then contour a slight offset of this volume before simplyfing it using QSlim.*

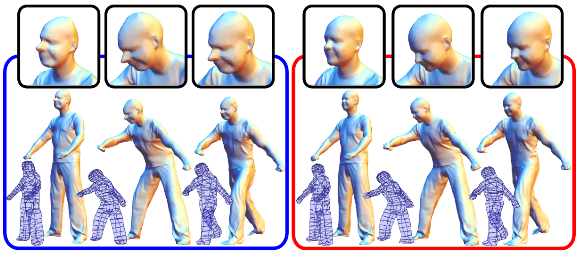


Figure 18: *Cage and model reconstruction comparison between a globally regularized over-determined least-squares (blue) and our approach (red), with the same setting (MVC, Laplacian minimization, $\lambda = 1$).*

For the *Capoeira* and *Horse* sequences (Fig. 14), our approach respectively yields an average L_2 error of $0.12\ 10^{-2}$ and $0.19\ 10^{-2}$ against $0.47\ 10^{-2}$ and $0.56\ 10^{-2}$ in the case of the skeleton-based framework by deAguiar et al. [dATTHP08] (numbers from the original paper), which is more than a 50% error improvement. Although they are not directly comparable (different computers), the timings of our implementation are up to 50 times smaller than in [dATTHP08], both on a pre-processing and per-frame basis.

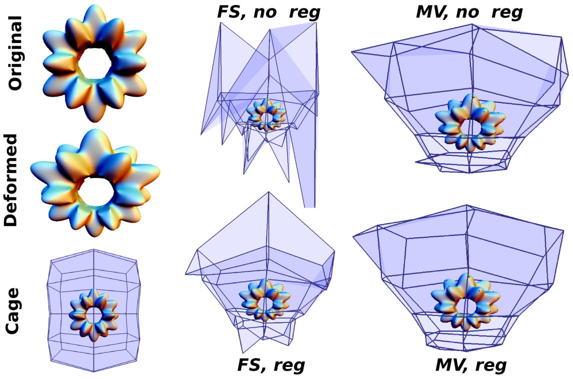


Figure 19: *Illustration of the technique on a* toy example. ***Left column:*** *Input (Original model, deformed model, and cage).* ***Middle column:*** *Results with full system (FS) inversion, without and with regularization.* ***Right column:*** *Results with MaxVol (MV) inversion, without and with regularization. In both cases, MVC were used, and the regularization was performed with parameters $\lambda = 1$, $s = 5$.*

INPUT SEQUENCE	POINT COUNT	CAGE (#V / #T)	FRAMES	PRE-PROCESS: MAXVOL + SVD (ms)	AVERAGE FRAME INVERSION (ms)	TOTAL (ms)
Fig. 14, 1st row	19,998	330 / 656	**499**	5,616	568	**289,048**
Fig. 14, 2nd row	19,998	330 / 656	**179**	5,616	568	**107,288**
Fig. 14, 3rd row	19,990	368 / 732	**437**	7,291	890	**396,221**
Fig. 14, 4th row	15,002	339 / 674	**169**	5,617	624	**111,073**
Fig. 14, 5th row	8,431	348 / 692	**48**	5,029	748	**40,933**

Table 1: *Detailed running times for Fig. 14. The pre-process, dominated by MaxVol, and the frame inversion timings are mostly dependent on the number of unknowns (number of cage vertices for MVC).*

Limitations As one can expect from a cage-based framework (error curves Fig. 14)), the maximum reconstruction error occurs on frames where the model pose is radically different (highly distorted) from the reference pose (e.g., 2nd row, middle). Another limitation is that, since our approach is cage-based, very small shape variations (e.g. small clothes motion) might be difficultly captured by the cage, since it tends to be designed for more global shape interactions.

6.4. Applications

Our high-level animated representations can be used for various applications.

Animation lossy compression: The reconstruction step of our algorithm only requires $\mathcal{P}_R$, $\mathcal{C}_R$ and the output cage positions across time. For instance, the first animation of Fig. 14 can be encoded in binary format with 115 MB (single connectivity, 499 embeddings), while it can be compressed down to 2.58 MB with our approach with a single connectivity, a single model embedding, a single cage connectivity and its 499 embeddings. The reconstruction error can be tuned with the spectrum threshold. Since our output cages smoothly vary over time, their vertex trajectories could even be compressed using orthogonal schemes (e.g. wavelets) for higher compression rates.

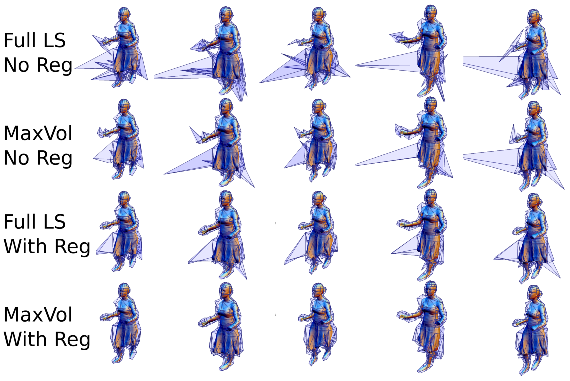


Figure 20: *Relative importance of the individual steps of our approach: inversion of the* Skirt *sequence with the over-determined least-squares (*Full LS No Reg*, first row), MaxVol (*MaxVol No Reg*, second row), the over-determined least-squares with spectral regularization (*Full LS With Reg*, third row, $s = 50$), MaxVol with spectral regularization (*MaxVol With Reg*, fourth row, $s = 50$).*

Animation transfer: Since they are clean and re-usable, the output cages can be employed for post-editing or animation transfer. In the **accompanying video**, an artist fits a model with complex topology (many handles) into the Capoeira reference cage. The cage sequence reconstructed with our algorithm automatically yields a smooth animation transfer to the new model.

Speeding up time-space processing tasks: Time-consuming processing tasks on time-varying 3D shapes can be drastically speeded up by considering the cages reconstructed with our algorithm, which have a memory footprint orders of magnitude smaller than the input sequences. In the **accompanying video** we present an application to the interactive exploration of shape spaces [WDAH10] (implementation provided by the authors). Given a set of model poses and a cage, our algorithm automatically adapts the cage to the entire set of poses. Then shape space exploration occurs interactively (500x speed-up) on the space of cages, delegating detail preservation to the underlying coordinate system.

7. Conclusion

We have presented CAGER, an automatic algorithm for the compact and stable encoding of animated 3D shapes into cage-based representations. The main contributions are an optimal selection of handles for cage coordinate inversion and a novel spectral regularization scheme, which localizes its *destructive* effects and automatically focuses on the most unstable cage vertices. Our technique is fast (GPU implementation) and generates smoothly varying, re-usable cages which still reconstruct the input sequence with higher accuracy than previous methods. We demonstrated the benefits of our algorithm for several 3D+t processing tasks.

Interestingly, our approach makes only few geometrical considerations. Our solution naturally rises from a thorough algebraic analysis of the inverse problem. Hence, our framework completely delegates the geometrical aspects to the chosen cage coordinate model, guaranteeing the generality and the versatility of the technique. In future work, we would like to explore the usability of our algebraic solution to other geometry processing tasks involving over-determined inverse problems. Also, an interesting research avenue is the automatic optimization of the cage geometry and connectivity in order to optimize the reconstruction of a target sequence, in other words, *caging by example*.

Acknowledgments

The captured performance data were provided courtesy of the research group 3D Video and Vision-based Graphics of the Max-Planck-Center for Visual Computing and Communication (MPI Informatik / Stanford). Other data-sets are courtesy of AIM@SHAPE. The authors would like to thank Tim Winkler for sharing his implementation of the paper [WDAH10] and the anonymous reviewers for their thoughtful remarks and suggestions. This work has been partially funded by the REVERIE European project, the 3DLife European Network of Excellence and the ANR iSpace&Time project.

References

[BCWG09a] Ben-Chen M., Weber O., Gotsman C.: Spatial deformation transfer. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), ACM, pp. 67–74. 3

[BCWG09b] Ben-Chen M., Weber O., Gotsman C.: Variational harmonic maps for space deformation. *ACM Trans. Graph. (SIGGRAPH)* (2009), 1–11. 2, 3, 7

[BK03] Botsch M., Kobbelt L.: Multiresolution surface representation based on displacement volumes. *Comp. Graph. Forum (EUROGRAPHICS) 22* (2003), 483–491. 7

[BS08] Botsch M., Sorkine O.: On linear variational surface deformation methods. *IEEE Trans. on Vis. and Comp. Grap.* (2008), 213–230. 2

[CMI07] Civril A., Magdon-Ismail M.: Finding maximum Volume sub-matrices of a matrix. *RPI Comp Sci Dept TR* (2007), 1–13. 4

[dAST*08] de Aguiar E., Stoll C., Theobalt C., Ahmed N., Seidel H.: Performance capture from sparse multi-view video. *ACM Trans. Graph. (SIGGRAPH) 27* (2008). 1, 8

[dATTHP08] de Aguiar E., Theobalt C., Thrun S., H.-P. S.: Automatic conversion of mesh animations into skeleton-based animations. *Comp. Graph. Forum (EUROGRAPHICS) 27* (2008). 1, 2, 10

[DSP06] Der K. G., Sumner R. W., Popovic J.: Inverse kinematics for reduced deformable models. *ACM Trans. Graph. (SIGGRAPH) 25* (2006), 1174–1179. 2

[ĒMI09] Ēivril A., Magdon-Ismail M.: On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science 410*, 47-49 (2009), 4801–4811. 4

[ESP08] Eigensatz M., Sumner R., Pauly M.: Curvature-domain shape processing. *Comp. Graph. Forum (EUROGRAPHICS) 27* (2008), 241–250. 7

[Flo03] Floater M. S.: Mean value coordinates. *Comp. Aided Geom. Design 20* (2003), 19–27. 2

[GOS*10] Goreinov S., Oaeledets L., Savostyanov D., Tyrtyshnikov E., Zamarashkin N.: How to find a good submatrix. *Matrix Methods: Theory, Algorithms and Applications* (2010), 247. 2, 4, 5

[HSL*06] Huang J., Shi X., Liu X., Zhou K., Wei L.-Y., Teng S.-H., Bao H., Guo B., Shum H.-Y.: Subspace gradient domain mesh deformation. *ACM Trans. Graph. (SIGGRAPH) 25* (2006), 1126–1134. 2

[HSS92] Hansen P., Sekii T., Shibahashi H.: The modified truncated svd method for regularization in general form. *SIAM Journal on Scientific and Statistical Computing 13* (1992), 1142. 7

[JMD*07] Joshi P., Meyer M., DeRose T., Green B., Sanocki T.: Harmonic coordinates for character articulation. *ACM Trans. Graph. (SIGGRAPH) 26* (2007). 1, 2, 4, 6

[JS11] Jacobson A., Sorkine O.: Stretchable and twistable bones for skeletal shape deformation. *ACM Trans. Graph. (SIGGRAPH ASIA)* (2011). 2

[JSW05] Ju T., Schaefer S., Warren J.: Mean value coordinates for closed triangular meshes. *ACM Trans. Graph. (SIGGRAPH) 24*, 3 (2005), 561–566. 1, 2, 4

[JT05] James D., Twigg C.: Skinning mesh animations. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 399–407. 3

[KG06] Kircher S., Garland M.: Editing arbitrarily deforming surface animations. *ACM Trans. Graph. (SIGGRAPH) 25* (2006), 1098–1107. 2

[KG08] Kircher S., Garland M.: Free-form motion processing. *ACM Trans. Graph. 27* (2008). 2

[KSO10] Kavan L., Sloan P., O'Sullivan C.: Fast and efficient skinning of animated meshes. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 327–336. 3

[LLCO08] Lipman Y., Levin D., Cohen-Or D.: Green coordinates. *ACM Trans. Graph. (SIGGRAPH) 27*, 3 (2008), 1–10. 1, 2, 4

[LSCO*04] Lipman Y., Sorkine O., Cohen-Or D., Levin D., Rossl D., Seidel H.-P.: Differential coordinates for interactive mesh editing. In *IEEE Shape Modeling International* (2004), pp. 181–190. 2

[LSLCO05] Lipman Y., Sorkine O., Levin D., Cohen-Or D.: Linear rotation-invariant coordinates for meshes. *ACM Trans. Graph. (SIGGRAPH) 24* (2005), 479–487. 2

[LWC06] Lee T.-Y., Wang Y.-S., Chen T.-G.: Segmenting a deforming mesh into near-rigid components. *The Visual Computer (Pacific Graphics) 22* (2006), 729–739. 2

[PSDB*10] Popa T., South-Dickinson I., Bradley D., Sheffer A., Heidrich W.: Globally consistent space-time reconstruction. *Comp. Graph. Forum (SGP) 29* (2010), 1633–1642. 1

[SA07] Sorkine O., Alexa M.: As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing* (2007), pp. 109–116. 2

[SC07] Schaefer S., C. Y.: Example-based skeleton extraction. In *Symposium on Geometry Processing* (2007), pp. 153–162. 2

[SF10] Savoye Y., Franco J.-S.: Cage-based tracking for performance animation. In *Asian Conference on Computer Vision* (2010). 3, 7

[SP04] Sumner R., Popović J.: Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 399–405. 3

[SZGP05] Sumner R. W., Zwicker M., Gotsman C., Popovic J.: Mesh-based inverse kinematics. *ACM Trans. Graph. (SIGGRAPH) 24* (2005), 488–495. 2

[TTB11] Thiery J.-M., Tierny J., Boubekeur T.: *Jacobians and Hessians of Mean Value Coordinates for Closed Triangular Meshes*. Tech. rep., CNRS LTCI Telecom ParisTech, 2011. 7

[WDAH10] Winkler T., Drieseberg J., Alexa M., Hormann K.: Multi-scale geometry interpolation. *Comp. Graph. Forum (EUROGRAPHICS) 29* (2010), 309–318. 11, 12

[WSLG07] Weber O., Sorkine O., Lipman Y., Gotsman C.: Context-aware skeletal shape deformation. *Comp. Graph. Forum (EUROGRAPHICS)* (2007), 265–274. 2

[XZY*07] Xu W., Zhou K., Yu Y., Tan Q., Peng Q., Guo B.: Gradient domain editing of deforming sequences. *ACM Trans. Graph. (SIGGRAPH) 26* (2007). 2, 3

[YBS07] Yoshizawa S., Belyaev A., Seidel H.: Skeleton-based variational mesh deformations. *Comp. Graph. Forum (EUROGRAPHICS)* (2007), 255–264. 2