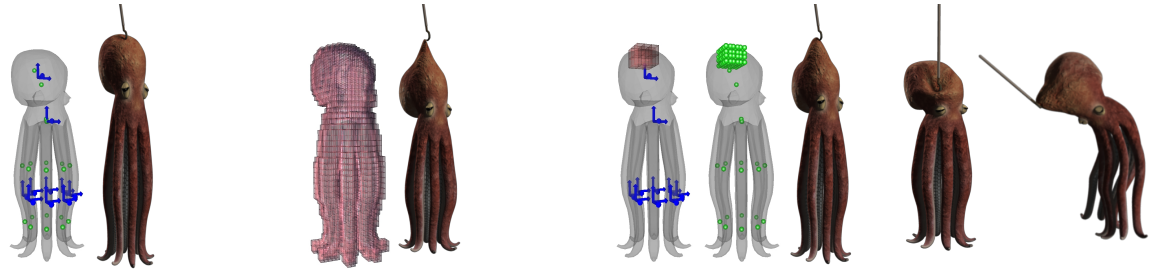
Multifarious Hierarchies of Mechanical Models for Artist Assigned Levels-of-Detail

Richard Malgat [1,2,6] Benjamin Gilles [3,1] David I.W. Levin [4] Matthieu Nesme [1,2] François Faure [5,2,1]

[1] INRIA [2] LJK-CNRS [3] LIRMM-CNRS [4] Disney Research [5] Univ. Grenoble [6] RDP-CNRS



(a) *Coarse frame-based simulation*

(b) *Fine FEM simulation*

(c) *Mixing a coarse frame-based simulation to a local FEM patch with our method*

Figure 1: *Using a hook to catch a coarsely discretized frame-based octopus is challenging due to the lack of deformation at the contact point (1a). One could resolve this problem with a high-resolution set of finite elements (1b) but, at the expense of runtime performance and with the introduction of spurious secondary motion. Multifarious Hierarchies of Mechanical Models allow an artist to add arbitrarily located detail simulations to the underlying coarse deformation models in order to produce the desired result. (1c).*

Abstract

We present a new framework for artist driven level of detail in solid simulations. Simulated objects are simultaneously embedded in several, separately designed deformation models with their own independent degrees of freedom. The models are ordered to apply their deformations hierarchically, and we enforce the uniqueness of the dynamics solutions using a novel kinetic filtering operator designed to ensure that each child only adds detail motion to its parent without introducing redundancies. This new approach allows artists to easily add fine-scale details without introducing unnecessary degrees-of-freedom to the simulation or resorting to complex geometric operations like anisotropic volume meshing. We illustrate the utility of our approach with several detail enriched simulation examples.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: Physically Based Animation, Deformable Solid, Multiscale Continuum Mechanics

1 Introduction

Physics-based animation has become an important arrow in the quiver of the visual effects practitioner – routinely used to produce creatures and scenes that appear plausible while simultaneously defying belief. In truth these effects depend on artists carefully sculpting their simulation results as much as they do on the simulations themselves. It is not uncommon for an artist to produce a highly scripted animation and then use a physical simulator to augment it with secondary motion (skin wrinkling, muscles flexing, metal bending). And, like a painter adding the finishing strokes to a canvas, a visual effects artist has an innate understanding of the scale, location and type of motion that should be added to the final scene.

Adapting the level-of-detail of a simulation is a well studied problem in mechanics. Currently this problem is addressed with specialized algorithms which can be divided into two classes: refinement approaches and coupled approaches. Refinement approaches alter the number of degrees of freedom (DOFs) in the simulation as a function of performance and accuracy. On the other hand, coupled approaches avoid refinement by using two (or more) simulation techniques which operate on different scales. These methods are common for turbulence simulation in fluids wherein a coarse fluid simulation is coupled (one-way or two-way) to a fine scale turbulence model. However, both refinement approaches and coupling approaches can be difficult for an artist to control as they can affect global changes in the simulation, requiring further parameter tuning to recapture underlying coarse motions. Unlike these previous approaches, what we seek is simulation paintbrush that an artist can use to augment kinematic or coarse dynamic animations with new details in an intuitive fashion.

In this paper we propose a new coupling approach for multi-scale, artist controllable simulation: the Multifarious Hierarchy of Mechanical Models (MHMMs). Multifarious Hierarchies are a general approach that allows a motion to be decomposed using a set of arbitrary, overlapping degrees-of-freedom. Analogous to the layers of popular image editing tools, MHMMs allow an artist to layer spatially varying simulations in order to carefully add secondary detail to coarse motions (Fig.1). Using a novel kinetic filtering approach, MHMMs avoid kinematic redundancy in the displacement discretization allowing for intuitive augmentation of the motion.

Furthermore, the geometry in the overlapping region experiences a hierarchically combined deformation field and this eases modeling since no geometrical operations are required to achieve the coupling. Finally, MHMMs increase computational efficiency because the total number of system degrees-of-freedom can be optimally chosen by the artist – no more than necessary will be used.

MHMMs allow an artist to produce robust, high-performance, multi-scale simulation results in an intuitive, layered fashion – yielding whole simulation results that are greater than a sum of their parts.

Contributions

The major contribution of this work is a general formulation for the layered addition of secondary motion to coarse animations. MHMMs allow for efficient, detailed simulations and in comparison to other methods, have the following advantages:

- *Controllable*: Users can add additional detail to a simulation at specified locations and at specified scales.
- *Spatially-Varying Layers*: Deformation models can be applied globally or locally allowing detail to be limited to appropriate regions of the domain (i.e. at contact points etc...).
- *Arbitrary Number of Levels*: Any number of overlapping layers can be used to craft the desired animation.
- *Layers of Varying Dimension and Material*: Combine rods, shells and volumetric models seamlessly; the detail can be used to represent a different material from the coarse model
- *Efficient*: MHMMs can be decomposed into independent systems that can be solved in parallel at runtime
- *General*: Any combination of degrees-of-freedom (particles, rigid and affine frames, modes...) or deformation models can be combined into a hierarchy
- *Fully-Coupled*: Forces and motions applied at one hierarchy level are propagated to the rest of the hierarchy

The remainder of our paper is organized as follows. Background is provided in sections 2 and 3, while our contributions are explained in sections 4 through 5. Section 6 is reserved for implementation details while results are presented in section 7.

2 Related work

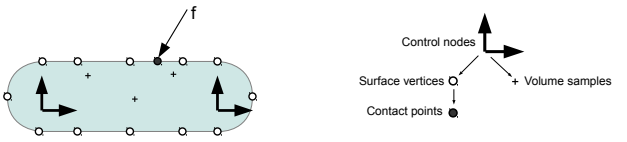
Adding relevant fine-scale details has long been an important area of research in graphics and engineering. As in the introduction we can divide previous methods into refinement approaches and coupling approaches. We begin by reviewing refinement approaches which can themselves be categorized by their degree of h-adaptivity and p-adaptivity. H- and p-adaptivity describe the ways in which simulation methods attempt to balance fidelity and performance. H-adaptive methods concern themselves with introducing new degrees-of-freedom by splitting individual elements while p-adaptive methods increase the degree of the polynomial approximation used for field variables. Both approaches have been well studied. In terms of h-adaptivity, Debunne et al [1999] introduce a multi-resolution, particle-based methodology for simulating linearly elastic objects. They provide a formalism for dynamically adding and removing particles in order to maintain a desired level of accuracy, and to locally switch between resolutions. Other h-adaptive remeshing approaches for Finite Element Simulation have been proposed, most recently in the context of thin shells, elasto-plastic simulation and mixed fluid solid simulations [Wicke et al. 2010; Narain et al. 2012; Clausen et al. 2013; Narain et al. 2013]. However, these techniques are limited to triangular and tetrahedral meshes which are unsuitable for many applications [Belytschko et al. 2000]. Tournier et al. [2014] introduced a h-adaptive approach for frame-based deformable solids allowing coarsening and refinement. Grinspun et al. [2002] present a h-hierarchical approach which reformulates remeshing as basis refinement. Their method is generally applicable to all element types but is limited to a single mechanical model and aligned levels of detail. This can require complicated mesh coarsening as a preprocessing stage. P-adaptive methods have been used with success to simulate fluids on coarse grids [Edwards and Bridson 2014] however for complicated domains such as solid objects, difficulties can arise. Because the polynomial basis is refined, p-adaptive methods begin with a coarse mesh that reasonably captures the geometry. As such, mesh generation can be problematic [Szabó et al. 2004].

Next we review coupling approaches which attempt to add detail to a simulation by "gluing" differing representations of an object together. These methods can be further divided into surface embedding methods, attachment methods and overlaying methods. Surface methods augment coarse volumetric simulations with high-resolution surfaces. Attachment methods provide a mechanism for coupling two different, non-overlapping simulations and overlaying methods use the superimposition of overlapping discretizations to add detail.

Surface embedding methods are popular in graphics and are used to retain reasonable surface detail for rendering while achieving fast simulation speed. Müller et al [2002] use barycentric coordinates to map a coarse deformation to a high resolution surface mesh. Similar approaches have been used to simulate viscoelastic objects with thin features [Wojtan and Turk 2008] as well as liquids [Wojtan et al. 2011]. Wojtan et al [2008; 2011] use the surface to track features but limit its influence on the underlying simulation (limited to computing surface tension forces). Some recent approaches procedurally deform the embedded surface, either from geometric analysis [Rohmer et al. 2010] or based on examples [Wang et al. 2010; Seiler and Harders 2012; Zurdo et al. 2013]. In these methods the fine detail cannot fully respond to user manipulation, relegating it to a more cosmetic role. In general these methods only deal with high-resolution detail at the surface of an object.

Attachment methods have also received a great deal of focus since Sifakis et al [2007] proposed a general methodology to connect models on mesh boundaries. Twigg et al proposed Point Cloud Glue [2010] to easily attach any sets of points using the Procrustes transform. Because simulations of differing resolutions can be attached, this method could be used to resolve details at certain parts of the model. However, attachment methods lack a mechanism for dealing with overlapping discretizations and so dynamic level-of-detail in a particular simulation domain would require an additional refinement algorithm. Furthermore, there are complications when allowing the detail level to be fully dynamic. Other methods based on substructuring are also commonly applied in engineering and graphics [Barbič and Zhao 2011; Kim and James 2012]. Often per-component reduced models are used to improve performance. Again, an additional algorithm for controlling level-of-detail within each substructure is required.

Overlapping methods have also been explored since the early work of Faloutsos et al [1997] where local Freeform deformation (FFD) lattices are embedded into larger global ones to allow for fine grained animator control. This method illustrates a particular instance of an MHMM and in our paper we focus on generalizing the concept to layerings of arbitrary discretizations. Two-way coupled algorithms were initially explored by Terzopoulos and collaborators [1988; 1990] using hierarchies of superquadrics imposed on rigid frames. More recently, Remillard et al. [2013] embed a high-resolution thin shell inside coarse FEM to resolve volume objects with stiff hulls. The mechanical coupling is perfomed us-



(a) *Discretization* **(b)** *Kinematic dependencies*

Figure 2: *Structure of a traditional model. Control nodes (here, moving frames), control surface vertices (empty circles) and volume sampling points (crosses). Contact forces are applied to surface points (filled circles) controlled by the surface vertices. (2a): Discretization with control nodes and sampling points. (2b): Kinematic dependencies.*

ing position constraints specific to the behavior of the stiff hull. Harmon et al [2013] enrich a space of modal deformations (see Barbic et al [2005]) with analytically defined detail functions (for a point force applied to an elastic half-space) to increase the expressivity of the model. These methods are very much in line with the spirit of our work but are, again, limited to particular instances of MHMMs (rigid modes and deformable superquadrics, poking functions adding details over modal models). Finally, the Eulerian-on-Lagrangian method [Fan et al. 2013; Fan et al. 2014] allows the coupling of a fine scale Eulerian simulation to any coarse scale Lagrangian simulation. The method is general but limited to Eulerian-Lagrangian coupling. Additionally the Eulerian simulation must cover entire deformable objects, meaning that detail cannot be added locally thus negating any potential performance gains (i.e in Fan et al. [2014] entire groups of muscles are covered with simulation grids rather than one muscle being covered by many grids) . Conversely, the focus of MHMMs is to provide a principled, controllable, layered methodology for adding local detail to coarser simulations, something that none of the previous work in this area provide.

3 Embeddings

This section provides background on embeddings, introduces notations and motivates our approach. Embeddings are intuitive – if one is playing a game of darts, the darts themselves become embedded in the board after each toss. Moving the board then causes all the darts to move with the same rigid motion.

A mathematical embedding (often used for animation [Capell et al. 2002; Galoppo et al. 2006]) formalizes this idea that the motion of an embedded mathematical object (a 3D point, deformation gradient, etc..) is influenced by the movement of some encapsulating domain (such as a rigid transform, a per-vertex displacement field or an affine blending). More generally, any mapping from one set of DOFs to another, such as vibration modes to local strains, can be used to build an embedding. Embeddings are particularly useful for synchronizing the motion of the embedded object with that of the encapsulating one. In this paper we use the term **master** to refer to the DOFs of the encapsulating object (the dart board) and the term **slave** to refer to the DOFs of the embedded object (the darts). **Slave** DOFs are kinematically "slaved" to **master** DOFs, hence their name. Furthermore we use the term **coarse** to refer to **master** DOFs that parameterize a lower-resolution simulation layer while the terms **fine** or **detail** refer to **master** DOFs that parameterize higher-resolution, detail layers.

Let us consider a set of material points, with $\mathbf{X}$ the vector of all their material coordinates, constant over time. As the state vector of the object, $\mathbf{q}$ (containing the displacements of the master DOFs from their initial coordinates), changes over time t, so do the displacements of the embedded, or slave, points, $\mathbf{u}_e$. The mapping relation is:

$$\mathbf{u}_e(t) = \mathcal{J}(\mathbf{q}(t), \mathbf{X}) \tag{1}$$

For the sake of clarity, we will drop the dependence on t in subsequent notations. Slave DOFs may in turn be the masters of other sets of DOFs, forming a hierarchy with the root as only set of master DOFs (Fig.2). The velocities $\mathbf{v}_e$ at the embedded level are given by:

$$\mathbf{v}_e = \mathbf{J}\dot{\mathbf{q}} \tag{2}$$

where $\mathbf{J} = \frac{\partial \mathcal{J}}{\partial q}$ is the Jacobian matrix of the mapping. Note that positions and velocities are only propagated "top-down", from the master to the slave DOFs. The Jacobian matrix is generally rectangular, and so mapping the velocities in the opposite direction requires solving an inverse kinematics problem, which often has no unique solution, if any. Conversely, the forces and impulses are only propagated bottom-up. Let $\mathbf{f}_e$ be forces at the embedded level, and $\mathbf{f}_q$ the corresponding forces at the master level. The Principle of Virtual Work states that the power of the forces must be the same at the two levels: $\mathbf{v}_e^T \mathbf{f}_e = \dot{\mathbf{q}}^T \mathbf{f}_q$, for any value of $\mathbf{v}_e$. This, combined with Eq.2, implies that

$$\mathbf{f}_q = \mathbf{J}^T \mathbf{f}_e. \tag{3}$$

Mass and stiffness matrices can be transferred "bottom-up" (from slave to master DOFs). For instance, if the inertia of the object is defined as a matrix $\mathbf{M}_e$, the equivalent matrix at the top level is $\mathbf{M}_q = \mathbf{J}^T \mathbf{M}_e \mathbf{J}$. While the complete derivation of all the properties of embeddings is out of the scope of this paper, we note that the above relations between forces and displacements at different levels of the kinematic hierarchy hold for any set of generalized coordinates and forces.

4 Hierarchies of Embeddings

4.1 Principle

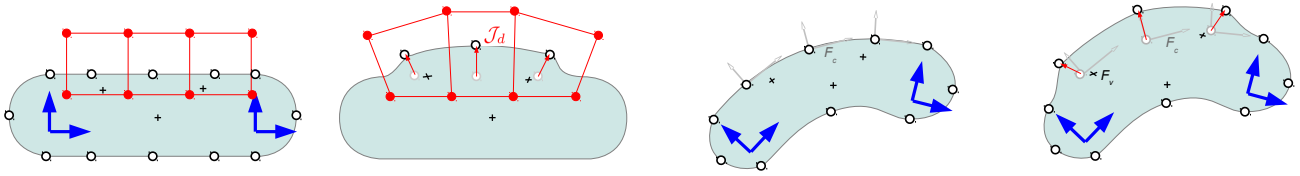
MHMMs consist of augmented embedding hierarchies. Rather than maintaining a strict master-slave relationship between embedded DOFs, we instead augment the embedding with its own master DOFs so that it can contribute new motion to the animation. One way to perform this augmentation would be to use a simple sum of embeddings $\mathbf{u}_e(t) = \mathcal{J}_1(\mathbf{q}_1(t), \mathbf{X}) + \mathcal{J}_2(\mathbf{q}_2(t), \mathbf{X})$. For small displacements this produces acceptable results, but its uncorrelated terms result in unacceptable artifacts as soon as large deformations or rotations occur.

In order to build an augmented embedding suitable for animating large deformations we instead modify Eq.1 by using a second embedding to change the material coordinates used by the first:

$$\mathbf{u}_e = \mathcal{J}_c(\mathbf{q}_c, \mathbf{X} + \mathcal{J}_d(\mathbf{q}_d, \mathbf{X})) \tag{4}$$

where $\mathcal{J}_c(\mathbf{q}_c, \mathbf{X})$ is a coarse displacement field and $\mathcal{J}_d(\mathbf{q}_d, \mathbf{X})$ is a new model used to add detail, as if a deformable object were embedded in the coarse model. In this approach, the detail changes the material coordinates used in the coarse model. We stress that such a hierarchy of embeddings can be superimposed on any mesh in order to parameterize its motion in a layered fashion.

Fig.3 shows the two-level case of an MHMM while Fig.4a illustrates the relationship between embeddings, DOFs and relevant physical objects (such as quadrature points). The top row of Fig.4a shows the relationship between the coarse DOFs (c) and a detail layer consisting of a local FEM simulation (d). All other objects



(a) *Rest state* **(b)** *FEM deformation only* **(c)** *Frame-based deformation only* **(d)** *Combined deformations*

Figure 3: *Example of a two-level MHMM, using a frame-based coarse model (blue) and an FEM detail model (red). The detail displacements $\mathcal{J}_d$ (3b) are applied in local frames defined by the deformation gradients $\mathcal{F}_c$ of the coarse displacements (3c) to generate the combined deformations (3d).*

are slave DOFS used for various purposes in the simulation algorithm. Surface vertices (s) are used for display and collision detection, contact points (p) for contact resolution while embedded particles (e) and volume samples (v) are used for numerical integration of relevant properties (i.e. mass and stiffness).

One potential problem with this approach is that combining embeddings using Eq.4 may require intense computations at each time step to update the coarse embedding based on the modified material coordinates, and may even introduce discontinuities when an embedded point moves from one cell of a mesh-based coarse model to another. To alleviate this problem, we express the detail motion in a local frame associated with each embedded point. This is essentially a first-order expansion of Eq.4:

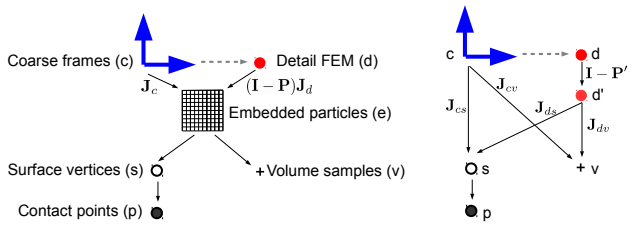
$$\mathbf{u}_e = \mathcal{J}_c(\mathbf{q}_c, \mathbf{X}) + \mathcal{F}_c \mathcal{J}_d(\mathbf{q}_d, \mathbf{X}) \tag{5}$$

where $\mathcal{F}_c$ is a block-diagonal matrix. For each embedded point i, the diagonal block represents the deformation gradient $\mathcal{F}_c^i = \frac{\partial \mathcal{J}_c^i}{\partial X^i} + \mathbf{I}$ of the coarse displacement at embedded point i. This non-orthonormal basis represents the rotation and deformation created by the coarse displacement, in which the detail displacement is applied, as illustrated by the example in Fig.3. Here, we assume that the coarse deformation field is locally affine (i.e.,$\mathcal{F}_c$ is uniform around $\mathbf{X}$), which is exact for some coarse models such as triangular or tetrahedral finite elements. In the general case, this assumption is reasonable when detail displacements are small or when coarse deformations are smooth. In contrast, summing embeddings requires small displacements of all models. Following Eq.2, the embedded velocity can be expressed as:

$$\mathbf{v}_e = \mathbf{J}\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_c & \mathbf{J}_d \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_c \\ \dot{\mathbf{q}}_d \end{bmatrix} \tag{6}$$

where the Jacobian is a sparse matrix with only two non-null blocks per embedded point: $J_c^i = [\frac{\partial \mathcal{J}_c^i}{\partial \mathbf{q}_c} + \frac{\partial \mathcal{F}_c^i}{\partial \mathbf{q}_c}\mathcal{J}_d^i]$, and $J_d^i = [\mathcal{F}_c^i \frac{\partial \mathcal{J}_d^i}{\partial \mathbf{q}_d}]$ for the coarse and detail models, respectively. However, it is easy to see that straightforwardly combining arbitrary embeddings using this method is not physically correct. Naïvely propagating a force from the embedded layer to both the coarse and detail layers (Eq.3) results in its duplication. Forces must be properly partitioned and then applied to each layer, ensuring that the total sum of work performed on the object is consistent.

Similar issues arise when dealing with other intrinsic object properties. Properties such as mass and stiffness are defined independently of the kinematic discretization either analytically or on a fine voxel map. These properties are sampled using embedded particles, as illustrated in Fig.4a. A given mass matrix $\mathbf{M}_e$ at the embedded level is not directly exploitable in Newton's law $\mathbf{f} = \mathbf{M}\mathbf{a}$, since the latter is only valid for independent DOFs, which requires us to modify



(a) *Basic* **(b)** *Optimized*

Figure 4: *Structure of the two-level MHMM shown in Fig.3.*

the mass matrix at a particular hiearchy level. Using the Jacobian of Eq.6, the equivalent mass matrix at parent level would be:

$$\mathbf{M} = \mathbf{J}^T\mathbf{M}_e\mathbf{J} = \begin{bmatrix} \mathbf{J}_c^T\mathbf{M}_e\mathbf{J}_c & \mathbf{J}_c^T\mathbf{M}_e\mathbf{J}_d \\ \mathbf{J}_d^T\mathbf{M}_e\mathbf{J}_c & \mathbf{J}_d^T\mathbf{M}_e\mathbf{J}_d \end{bmatrix} \tag{7}$$

Fan et al. [2013] observed that, while the diagonal blocks are symmetric-positive-definite (SPD), this mass matrix is generally singular, due to the off-diagonal coupling blocks. This happens in the case of redundancies between the coarse and detail embeddings. For instance, if the coarse field is a rigid displacement and the detail is an FEM mesh, then a translation of the embedded object can be obtained using a translation of the frame as well as a uniform displacement of the FEM nodes. To alleviate the singularity, they use a least-squares approach to transfer fine scale Eulerian displacements onto a coarse Lagrangian discretization. In contrast to this, we propose a more physically-based approach, using the mass matrix, as explained in the next section.

4.2 Kinetic Filtering

We attempt to find the corresponding generalized velocities at the coarse level, $\dot{\mathbf{q}}_c$, that best match $\mathbf{v}_e$, the velocity at the embedded level. This can be done by solving a mass-weighted least squares problem, the solution of which yields $\dot{\mathbf{q}}_c = \mathbf{M}_c^{-1}\mathbf{J}_c^T\mathbf{M}_e\mathbf{v}_e$, where $\mathbf{M}_c = \mathbf{J}_c^T\mathbf{M}_e\mathbf{J}_c$. Note that, if one takes $\mathbf{M}_e$ to be a multiple of identity, this reduces to the classical pseudo inverse solution to a system of equations. Now we can define $\mathbf{v}'_e$ the embedded velocity captured by $\dot{\mathbf{q}}_c$ as

$$\mathbf{v}_e^c = \mathbf{P}\mathbf{v}_e \tag{8}$$

$$\mathbf{P} = \mathbf{J}_c\mathbf{M}_c^{-1}\mathbf{J}_c^T\mathbf{M}_e. \tag{9}$$

We can also compute the remaining detail level velocity as

$$\mathbf{v}_e^d = (\mathbf{I} - \mathbf{P})\,\mathbf{v}_e \tag{10}$$

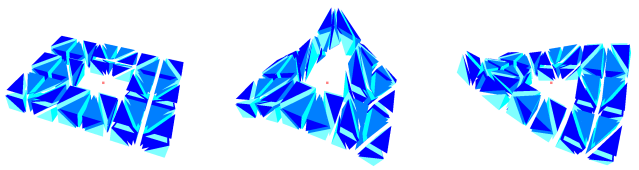


Figure 5: *Filtered detail motion. Left: rest shape. Middle, right: detail deformations generated by external forces.*

where $\mathbf{I}$ is the identity matrix. This leads us to define the complementary operator $\bar{\mathbf{P}} = \mathbf{I} - \mathbf{P}$ which computes detail velocity (at the embedded level) from the total velocity. It is straightforward to see that $\mathbf{p}_e^c = \mathbf{M}_e \bar{\mathbf{P}} \mathbf{v}_e$ yields an embedded momentum in the nullspace of $\mathbf{J}_c^T$. Concretely, if the coarse layer is a rigid transform and the fine layer is a finite element simulation, $\bar{\mathbf{P}}$ strips rigid motion out of the FEM velocity field while maintaining the non-rigid motions. This avoids kinematic redundancy since rigid motions can only be manifested at the coarse layer.

In order to continue, we define the Jacobian of our system as

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_c & \bar{\mathbf{P}}\mathbf{J}_d \end{bmatrix} \tag{11}$$

and consequently the momentum computed by the off-diagonal blocks of the generalized mass matrix for an arbitrary detail velocity $\dot{\mathbf{q}}_d^*$ is

$$\mathbf{J}_c^T \mathbf{M}_e \bar{\mathbf{P}} \mathbf{J}_d \dot{\mathbf{q}}_d^* = \mathbf{J}_c^T \mathbf{p}_e^* = \mathbf{0}. \tag{12}$$

Because $\dot{\mathbf{q}}_d^*$ was chosen arbitrarily, this is equivalent to the off-diagonal mass matrix block, $\mathbf{J}_c^T \mathbf{M}_e \bar{\mathbf{P}} \mathbf{J}_d$, being $\mathbf{0}$. Note that this does not occur when using the non-mass-weighted formulation of Fan et al. [2013] and that later, we will exploit this property to evaluate hierarchy levels in parallel.

The resulting form of Newton's equation is:

$$\begin{bmatrix} \mathbf{J}_c^T \mathbf{M}_e \mathbf{J}_c & \\ & \mathbf{J}_d^T \bar{\mathbf{P}}^T \mathbf{M}_e \bar{\mathbf{P}} \mathbf{J}_d \end{bmatrix} \begin{bmatrix} \mathbf{a}_c \\ \mathbf{a}_d \end{bmatrix} = \begin{bmatrix} \mathbf{J}_c^T \mathbf{f}_e \\ \mathbf{J}_d^T \bar{\mathbf{P}}^T \mathbf{f}_e \end{bmatrix} \tag{13}$$

The upper-left block is SPD provided that $\mathbf{M}_e$ is symmetric and $\mathbf{J}_c$ is not rank-deficient. While the lower-right block is singular due to the filter, it can be solved using a Filtered Conjugate Method [Ascher and Boxerman 2003].

An example of tetrahedral FEM detail combined with a rigid coarse motion is shown in Fig.5. The rigid frame is fixed for illustration purposes, so that only detail deformations are shown. An external force applied to a single point creates deformations. The force is split by the transposed Jacobian, and the FEM model gets a force distribution which can only generate null linear and angular momenta. It can thus generate any displacement but a rigid one. Notice that there is no fixed point in our detail model. Contrary to previous approaches, the connection between the two models is made by the filtered Jacobian, not by attachment points. This allows us to straightforwardly combine linear modal deformation modes as detail on top of rigid motion, as shown in Section 7.4, to avoid the computation of nonlinear deformation modes.

4.3 Forces

Our framework is compatible with all the usual material laws and contact behaviors. In the simple approach illustrated in Fig.4a, the surface layer s, as well as the deformation energy integration samples (Gauss points in layer v), are embedded in layer e using interpolation. Embedding deformation gradients in layer v allows us to implement all the standard strain measures (such as corotational and Green-Lagrange), and associated constitutive laws (such as Hookean or hyperelastic). The associated generalized forces are stresses, which are propagated as forces to layer e, which also collects the contact forces mapped from p through s. These forces are split between coarse and detail forces as explained in Section 4.2.

4.4 N-levels hierarchies

Kinetic filtering can be straightforwardly extended to handle more than two levels in the hierarchy. Let $\mathbf{q}_i$, $\mathcal{J}_i(t)(\mathbf{q}_i, \mathbf{X})$ and $\mathcal{F}_{i-1} = \frac{\partial \mathcal{J}_{i-1}}{\partial \mathbf{X}_{i-1}}$ respectively the coordinates of the DOFs, the detail and the local frame at level i. Each detail layer is applied in the parent frame, so that for N levels, we get:

$$\begin{aligned} \mathbf{u}_e &= \mathcal{J}_1 + \mathcal{F}_1\mathcal{J}_2 + \mathcal{F}_1\mathcal{F}_2\mathcal{J}_3 + \dots \\ &= \mathcal{J}_1 + \textstyle\sum_{k=2}^{N} (\prod_{j=1}^{k-1} \mathcal{F}_j)\mathcal{J}_k \end{aligned} \tag{14}$$

The corresponding Jacobian is:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \dots & \bar{\mathbf{P}}_{i-1}\mathbf{J}_i & \dots \end{bmatrix} \tag{15}$$

$$\bar{\mathbf{P}}_0 = \mathbf{I}, \quad \bar{\mathbf{P}}_i = \bar{\mathbf{P}}_{i-1}.(\mathbf{I} - \mathbf{P}_i) \tag{16}$$

$$\mathbf{P}_i = \bar{\mathbf{P}}_{i-1}\mathbf{J}_i \mathbf{M}_{ii}^{-1} \mathbf{J}_i^T \bar{\mathbf{P}}_{i-1}^T \mathbf{M}_e \bar{\mathbf{P}}_{i-1} \tag{17}$$

$$\mathbf{M}_{ii} = \mathbf{J}_i^T \bar{\mathbf{P}}_{i-1}^T \mathbf{M}_e \bar{\mathbf{P}}_{i-1} \mathbf{J}_i \tag{18}$$

This formulation achieves decoupling since off-diagonal blocks of the mass matrix are null (see proof in Appendix A).

5 Optimizing Performance

Constructing the kinetic filter requires relating momenta of one level of the hierarchy to the momenta of another. In the naïve implementation described above we do this by storing high resolution information on the embedded particles in layer e (the density of which must be sufficient to represent the deformation field at the most detailed layer, d, in the range of the detail deformation). Thus the matrix $\mathbf{P}$, used by the kinetic filter, is dense and is of dimension of the number of particles in the embedded layer. While $\mathbf{P}$ is never inverted, this may result in increased computation times due to the cost of performing dense matrix-matrix multiplication to form the per-layer mass matrices (the size of these is strictly determined by the number of master DOFs in the particular layer). In order to reduce the computation time without modifying the sampling scheme, we present an optimization to bypass the embedded layer e completely, thus directly connecting v and s to the master layer c-d. Each of the points in layers s and v have their own material coordinates and can thus be directly embedded into c-d.

For layer v, we compute deformation gradients $\mathcal{F}_v$ by spatially differentiating the embedding of Eq.5 with respect to the material coordinates of the sampling points:

$$\mathcal{F}_v = \mathcal{F}_c \mathcal{F}_d \tag{19}$$

This corresponds to the composition of the coarse and detail deformation gradients as shown in Fig.3. The stresses $\mathbf{f}_v$ are computed using the deformation gradients and the local material constitutive law, and integrated in space using standard cubature. Stresses could be directly mapped up as forces on layers c and d based on the transposed Jacobian matrix (Eq.3). However, the question is how to filter the Jacobian of this embedding, to avoid "double counting" stresses while converting them to forces at the master level.

Let $\mathbf{J}_{cv}$ be the Jacobian of the embedding of deformation gradients in the coarse model, and $\mathbf{J}_{dv}$ be the *unfiltered* Jacobian in the detail model, obtained by differentiation of Eq.19 with respect to the coordinates of the DOFs. We convert stresses to filtered forces in the following way:

$$\mathbf{f}_d = \mathbf{J}_d^T \bar{\mathbf{P}}^T \mathbf{M}_e [\mathbf{J}_c \mathbf{M}_c^{-1} \mathbf{J}_{cv}^T + \bar{\mathbf{P}} \mathbf{J}_d \mathbf{M}_d^{-1} \mathbf{J}_{dv}^T] \mathbf{f}_v \quad (20)$$

The above equation transforms the force $\mathbf{f}_v$ from v to d using the following steps (right-to-left):

1. Transform forces from v to c and d using $\mathbf{J}_{cv}^T$ and $\mathbf{J}_{dv}^T$
2. Convert forces at d and c to accelerations using $\mathbf{M}_d^{-1}$ and $\mathbf{M}_c^{-1}$
3. Sum contributions to the corresponding embedded accelerations using $\mathbf{J}_c$ and $\mathbf{J}_d$
4. Compute the force at the embedded level using $\mathbf{M}_e$
5. Split the force on the master layer using kinetic filtering

By re-arranging the above equation using Eq.9, we get:

$$\mathbf{f}_d = (\mathbf{I} - \mathbf{P'}^T) \mathbf{J}_{dv}^T \mathbf{f}_v \quad (21)$$

$$\mathbf{P'} = \mathbf{M}_d^{-1} \mathbf{J}_d^T \mathbf{M}_e \mathbf{P} \mathbf{J}_d \quad (22)$$

$$\mathbf{J'} = \begin{bmatrix} \mathbf{J}_{cv} & \mathbf{J}_{dv}(\mathbf{I} - \mathbf{P'}) \end{bmatrix} \quad (23)$$

Note that now the dimension of our kinetic filter, $\mathbf{P'}$ are dictated by the number of DOFs in the detail level only, a signfigant reduction in size that yields a corresponding performance increase

The square matrix $(\mathbf{I} - \mathbf{P'}^T)$ acts as a filter on the detail forces, while its transpose acts as a filter on the detail velocities. Moreover, since the above formulation holds for any other slave layers such as s, it can be applied once to the force on d and then accumulated bottom up, as illustrated using d' in Fig.4b.

5.1 Decoupled implicit integration

For implicit time integrators we can gain further performance improvements by modifying our kinetic filter so that it decouples hierarchy levels during the implicit solve. This allows us to parallelize our velocity update steps on a per-layer basis. Consider the Implicit Euler method, which requires repeated solution of the following equation:

$$\left(\mathbf{M} - h^2 \mathbf{K}\right) a = f + h\mathbf{K}\mathbf{v} \quad (24)$$

where $\mathbf{K} = \frac{\partial f}{\partial q}$ is the stiffness matrix, and h is the time step. Given the mass matrix $\mathbf{M}_e$ at the embedded level, the mass matrix at the master level is computed as explained in Section 4.2 using the filtered Jacobian of Eq.11. We do not assemble the stiffness matrix at the master level, since we use a Conjugate Gradient solver and propagations to compute the matrix-vector products. Let $\mathbf{H} = \left(\mathbf{M} - h^2 \mathbf{K}\right)$ be the integration matrix. The filtering method to cancel the off-diagonal blocks of the mass matrix presented in Section 4.2 can straightforwardly be extended to this matrix by replacing the projection $\mathbf{P}$ of Eq.8 with:

$$\mathbf{v''} = \mathbf{S}\mathbf{v} = \mathbf{J}_c \mathbf{H}_c^{-1} \mathbf{J}_c^T \mathbf{H}_e \mathbf{v}, \quad (25)$$

Equation 24 becomes:

$$\begin{bmatrix} \mathbf{J}_c^T \mathbf{H}_e \mathbf{J}_c & \\ & \mathbf{J}_d^T (\mathbf{I} - \mathbf{S}^T) \mathbf{H}_e (\mathbf{I} - \mathbf{S}) \mathbf{J}_d \end{bmatrix} \begin{bmatrix} \mathbf{a}_c \\ \mathbf{a}_d \end{bmatrix} = \begin{bmatrix} \mathbf{J}_c^T (\mathbf{f}_e + h\mathbf{K}_e \mathbf{v}) \\ \mathbf{J}_d^T (\mathbf{I} - \mathbf{S}^T)(\mathbf{f}_e + h\mathbf{K}_e \mathbf{v}) \end{bmatrix} \quad (26)$$

This results in two systems coupled only through the right-hand term. These can be solved independently, possibly using specialized solvers or in parallel. Our coupling is neither fully explicit nor fully implicit. The matrix $\mathbf{S}$ is used to implicitly account for the change in coupling force due to changes in the detail displacements. However, it ignores changes in the kinetic filter and $\mathbf{J}_d$. This is analogous to standard implementations of co-rotational FEM, which assume that the rotation matrix is fixed across time steps. In practice we've found this integration scheme to be stable for large time steps. An example is shown in Section 7.4 wherein the rigid dynamics are handled by a generic solver while the modal model is handled using a specialized, preconditioned conjugate gradient solver described in the following sections.

5.2 Pre-factored detail systems

In MHMMs, detail motions are expressed in local frames relative to the coarse motion ($\mathbf{J}_d^T$ in the right hand side of eq 26 transforms filtered forces to the detail coordinate system, where the detail system is solved). The nature of the detail motion is that it is high-frequency, typically with lower amplitude - thus we can apply small deformation simplifications. By pre-factoring the lower right term of the matrix of Eq. 26, supposing that $\mathcal{J}_d$ is small enough, we can increase computational time, at the cost of accuracy. Examples of such systems are shown using linear FEM(Figure 7), and linear modal simulations(Figure 10).

For non-interactive scenarios (e.g. animations for films) this prefactoring has no drawbacks. However, in applications wherein the user manipulates the mesh by applying an external force, the prefactorization prevents us from implicitly integrating this interaction force. It is worth nothing that this limitation is common to most methods that prefactor the implicit integration system matrix.

5.3 Preconditioning Nonlinear Problems

Our prefactored, per-layer, system matrices can also be used to accelerate the solution of non-linear detail simulations by preconditionning the detail solver using the initial prefactored system matrix (see Figure 8): Performance improvements of up to 50% have been observed in our experiments. This greatly reduces the cost of adding extra detail to *any* simulation.

5.4 Precomputation Cost

Our precomputation requires the assembly of projection matrices (involving the inversion of mass matrices as in standard direct solvers, and a matrix multiplication of the size of the embedding at most) and the proposed prefactorization (cholesky factorization on a matrix of the layer size). Practically speaking, precomputation time is on the order of a few simulation timesteps at most.

6 Implementation Details

In this section we review some relevant implementation details as well as expound upon important optimizations that can be applied to MHMMs.

At initialization time, we first organize the model based on the property map of the object, as illustrated in Fig.4a. Computing the mass $\mathbf{M}_e$ at the embedded level and storing it as a constant in the master layer is a reasonable and efficient approximation. Based on the stiffness distribution and the embedding, we compute the volume samples of layer v. We then set up the direct embeddings and remove the embedded layer e as explained in Section 5 and illustrated in Fig.4b. Since gravity generates translational forces, its contribution to the detail is null when translation is kinematically feasible by the coarse model. In this case, we apply it directly at the coarse

level. Otherwise, it is possible to split the weight across volume sampling points.

In general, the Jacobians change over time and must be updated at each step of the simulation, and the filters as well. However, many popular blendings such a linear FEM, linear modes, or affine frame-based blendings have constant Jacobians, and result in constant $\partial \mathcal{J}_c / \partial \mathbf{q}_c$ or $\partial \mathcal{J}_d / \partial \mathbf{q}_d$ matrices which allow us to save computation time.

Assuming that the fine displacement of the detail $\mathcal{J}_d$ is small, the coarse Jacobian matrix $\mathbf{J}_c$ and filter $\mathbf{P}$ can be considered constant. When the coarse displacement is large, the change of local frames $\mathcal{F}_c$ in Eq.5 requires updates of $\mathbf{J}_d$.

Lazy updates of Jacobians can lead to poor filtering performance and this manifests as coarse motion bleeding into the detail layers. This is due to violating the constraint that off-diagonal blocks of the mass matrix should be zero. In practice we correct this using standard stabilization techniques. Such stabilization may be achieved by fixing the edges of the detail model, which also ensures continuity in the displacement, at the interface of this model. With stabilization in place the method is robust enough so that, in practice, any linear solver like CG or MINRES will work for every scenario.

Pre-computing the filter once for the whole simulation does not lead to visual artifacts (as can be seen in the accompanying video) while recomputing the filter doubles the computation time of a given time step.

7 Results

In this section, we exhibit the versatility of our approach using different combinations of deformation models (2d/3d finite elements, rigid, frame-based, modal subspace) in conjunction with artist positioned secondary motion.

7.1 Physical Plausibility

To show that our algorithm produces physically plausible results we compared it to a high-resolution FEM simulation of a beam undergoing 3-point bending. Figure 6 shows both the FEM result and the result of an MHMM composed of two coarse hexahedra, manually overlaid with a fine hexahedral grid. The relative displacement error is less than 20% and visually the results are indistinguishable.

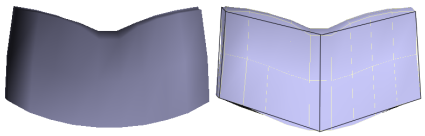


Figure 6: *Comparison on flexion. Left: standard FEM. Right: MHMM with coarse FEM + detail*

In the accompanying video, the simulation of a cantilever beam in contact shows a major advantage of MHMMs with regards to artist directed level-of-detail. By adding detail only in desired locations (near the contact point) MHMMs keep the global motion of the beam close to the input coarse motion. The fine FEM grid leads to spurious secondary motion that alters the beams trajectory significantly. Note that all the beams reach identical final configurations.

7.2 FEM on a rigid frame

An artist can choose to apply detail motion to a whole object, as shown in Fig.7. Here, a rigid model is enriched with tetrahedral FEM (red edges on Fig.7-left), covering the entire object. Note that there is no attachment point between the rigid and the FEM models.

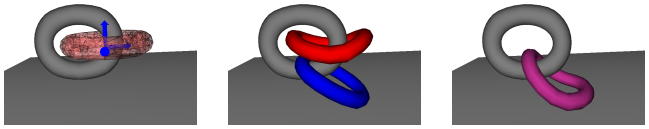


Figure 7: *Rigid frame (blue) + global FEM (red). Left : initial position. Middle : simulated models shown separately. Right : Combined result. Grey objects are fixed colliders.*

Figure 8 shows an example of a local FEM volumetric patch hierarchically attached to a moving frame. This simple simulation, which could also be produced using traditional coupled models, shows that our detail model can undergo large deformations. The detail model is fixed at its boundary with the rest of the object, to enforce the continuity of the displacements.

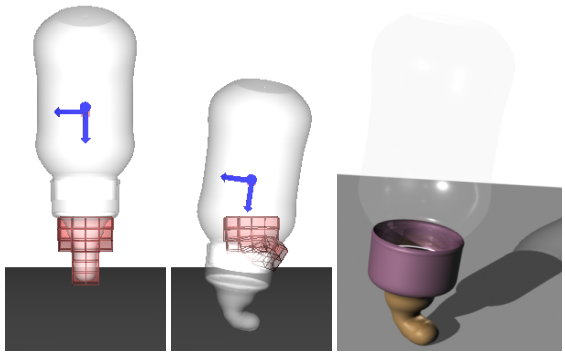


Figure 8: *Rigid frame + local FEM: a baby bottle with a deformable soother. The detail level FEM is displayed with its local coordinate system centered on the origin.*

Even for a very stiff material, the weight is correctly simulated in these two examples, because the kinetic filter maps it to the rigid model. This alleviates the well-known gravity artifact which occurs when iterative solvers are used with stiff FEM. Incomplete solutions lead to stiff objects falling more slowly.

7.3 FEM on frame-based deformation models

Our method allows an artist to add local secondary motion to any type of coarse simulation or kinematic animation. In Fig.1 the coarse model of the octopus is frame-based [Gilles et al. 2011], which allows for efficient simulation of the global motion using linear blend skinning. The artist then chooses to add a small, high-resolution FEM patch to the octopus' head in order to capture the deformation caused by its interaction with the hook.

Figure 9 shows a similar example where the artist has added a cylinder of high-resolution FEM to capture the pinching caused by the contacting string.

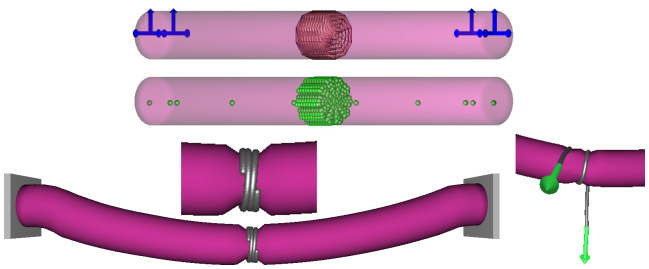


Figure 9: *A string strangulates a deformable cylinder (purple). A local tetrahedral FEM patch (red) allows a precise deformation in the contact area while coarse affine frames (blue) simulate the global deformation.*

Allowing artists to easily had detail where its needed has the added benefit of reducing computational time as well as offering a corresponding reduction in memory footprint (Table 1).

7.4 Non-linear Modal Subspace

Non-linear subspace deformation can be easily and efficiently achieved with our technique by combining a coarse frame-based deformation field and linear subspace deformation modes. These modes can be drawn from simulation data, capture data or hand crafted by an artist. The MHMM locally transforms each mode by the blended displacement of the frames. The well known artifacts due to linearity are reduced as the number of frames increases. This is akin to a co-rotational simulation method wherein the rotation is encoded by the frames rather than resulting from a polar decomposition.

Again we see that MHMMs allow for better control of animations. As shown in the accompanying video, the scripted displacement of one of the frame can be automatically enriched with simulated deformations (based on the remaining frames and modal degrees of freedom).

To exhibit particular visual features, a third level can be added. In Figure 10, local detailed deformations are achieved using a local FEM patch. In Fig.11, we augment a global, linear modal model using local subspace deformation modes which allows for fine tuning of the deformation degrees of freedom.

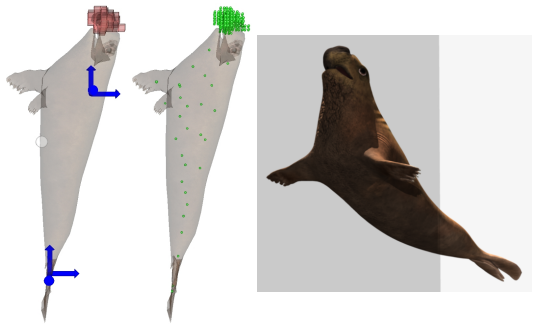


Figure 10: *A 3-levels hierarchy combining frame-based, linear modal subspace, and hexahedral finite elements models. Left: Frame and FEM discretization; Middle: volume samples; Right: Simulation.*

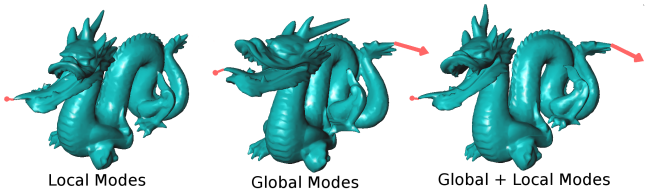


Figure 11: *Frame-based, global subspace and local subspace deformation models combined in a 3-level hierarchy.*

7.5 Multi-dimensional material

MHMMs also provide a way for artists to modify the material composition of simulated object. For instance, because our approach makes no assumption on the topology of the models, one can easily add a stiff skin to coarse soft tissue (Figure 12)

The coarse FE achieves volume preservation, while the fine surface creates folds. The coupling achieved by the kinetic filter generates the well-known buckling behavior without additional constraints.

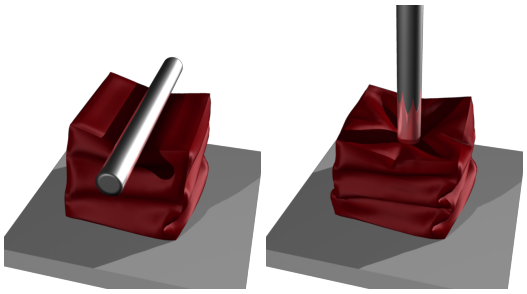


Figure 12: *A detailed thin shell embedded in a single coarse hexahedral element.*

In Fig.13, we simulate the pinching of a patch of skin with hypoderm using the epiderm as 2D detail on top of a two-material coarse volumetric grid.

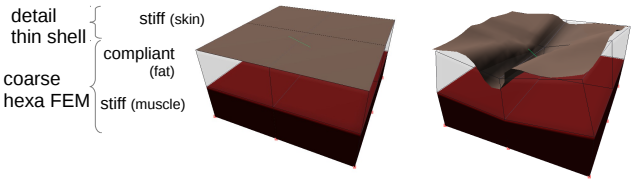


Figure 13: *Skin folding simulation using a coarse volumetric FEM mesh (in blue) combined with a thin shell patch and several material properties.*

7.6 Multilayer Editing of Animations

Here we show an example of multilayered animation editing, akin to the layered approach used in many popular image editing suites. Our kinetic filter helps ensure that adding new detail levels to a simulation will have a limited effect on any underlying gross motion. Figure 14 shows a 3 level hierarchy applied to an animated character. The coarse motion is prescribed by an artist using standard skinning techniques. The artist has then added a coarse FEM grid to capture the jiggling of the character's belly along with a finer, local grid to add the indent caused by the character's finger. Our approach can emulate Rig-Space physics [Hahn et al. 2012] using a hierarchy where the master layer is the rig that controls a fine tetrahedral FEM model. However, we improve upon this technique by allowing artists to add physical details without the need to modify the characfer rig itself. One could use a full high resolution simulation such as in McAdams et al. [2011], however using MHMMs requires far fewer degrees-of-freedom and allows for direct augmentation of the skinned surface animation (rather than being limited to having the bone motion drive a soft elastic character). All while permitting significantly more user control of deformation placement. Not to mention that our technique is compatible with a wider range of material models.

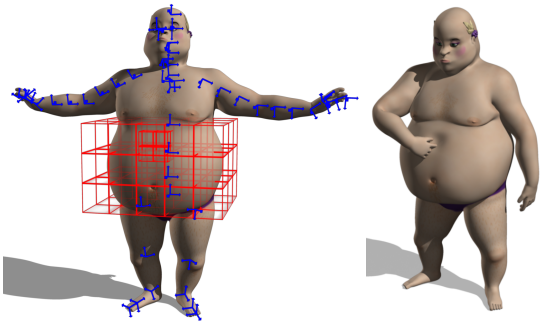


Figure 14: *A full body with a coarse rigid motion and three levels of hierarchy easily controlled by an artist.*

7.7 Performance

Frame rates for our examples are given in Table 1. For a fair comparison, we use the same solvers and contact handling mechanisms for all deformable models. MHMMs achieve large speedups compared to models with uniformly fine resolutions (corresponding to the detail we wish to simulate). As is the case with other techniques, the number of degrees of freedom impacts the performance of our method but, we gain invariance with respect to the spatial dimension of the detail. Therefore, for a given number of degrees of freedom, details at arbitrarily fine scales can be simulated in constant time. Other methods would require extremely dense discretizations, re-meshing or adaptive refinement to accomplish this.

8 Limitations and Future Work

The main bottleneck of MHMMs is potential coupling of all detail DOFs through the coarse layer. This results in a dense kinetic filter matrix, which slows down the required matrix products. In such cases, a more standard, high resolution simulator is likely to achieve better performance. For MHMMs, pre-factorization mitigates this problem, as the filter is only required to compute the right hand side vector. We suspect that filtering may be further accelerated using a multigrid approach, but defer this study to future work. Finally, the mechanical coupling between layers could be improved using the geometric stiffness [Tournier et al. 2015] that accounts for the variation of the Jacobian within timesteps.

9 Conclusion

Multifarious Hierarchical Mechanical Models are a new approach to solid simulation, based on hierarchically coupling deformable models. Using a novel operator, the kinetic filter we provide a single tool that can be used by artists to allow simple editing of a simulation as well as to simulate heterogeneous materials (see our skin example) or provide additional detail in standard simulations, even ones which are highly non-linear. As such, MHMMs bring the convenience of layered editing, common place in the image processing domain, to physically-based animation.

Our algorithm is implemented using the SOFA open-source library supported by INRIA and partners [Faure et al. 2012], and is publicly available in the *MultifariousHierarchies* plugin (`https://gforge.inria.fr/projects/mhmm`).

Acknowledgments

The authors wish to thank Laura Paiardini for her help with the models, textures, renderings and video editing and Romain Testylier for his help with the rod simulation. We also thank the anonymous reviewers for the suggestions for improving the paper. This work was supported by funding from French FUIs *Dynamit* and *Collodi*, French ANR project *SoHusim* and INRIA project *Morphogenetics*.

References

ASCHER, U. M., AND BOXERMAN, E. 2003. On the modified conjugate gradient method in cloth simulation. *The Visual Computer 19*, 7-8.

BARBIČ, J., AND JAMES, D. L. 2005. Real-time subspace integration for st. venant-kirchhoff deformable models. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 24.

BARBIČ, J., AND ZHAO, Y. 2011. Real-time large-deformation substructuring. *ACM Trans. Graph. 30*, 4 (July), 91:1–91:8.

BELYTSCHKO, T., LIU, W., AND MORAN, B. 2000. *Nonlinear finite elements for continua and structures*. Wiley.

CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. Interactive skeleton-driven dynamic deformations. *ACM Trans. Graph. 21*, 3 (July), 586–593.

CLAUSEN, P., WICKE, M., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2013. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Trans. Graph. 32*, 2 (Apr.), 17:1–17:15.

DEBUNNE, G., DESBRUN, M., BARR, A. H., AND CANI, M.-P. 1999. Interactive multiresolution animation of deformable models. In *Eurographics Workshop on Computer Animation and Simulation*.

EDWARDS, E., AND BRIDSON, R. 2014. Detailed water with coarse grids: Combining surface meshes and adaptive discontinuous galerkin. *ACM Trans. Graph. 33*, 4 (July), 136:1–136:9.

FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 1997. Dynamic free-form deformations for animation synthesis. *IEEE Transactions on Visualization and Computer Graphics 3*, 3.

FAN, Y., LITVEN, J., LEVIN, D. I. W., AND PAI, D. K. 2013. Eulerian-on-lagrangian simulation. *ACM Transactions on Graphics 32*, 3.

FAN, Y., LITVEN, J., AND PAI, D. K. 2014. Active volumetric musculoskeletal systems. *ACM Trans. Graph. 33*, 4 (July), 152:1–152:9.

FAURE, F., DURIEZ, C., DELINGETTE, H., ALLARD, J., GILLES, B., MARCHESSEAU, S., TALBOT, H., COURTECUISSE, H., BOUSQUET, G., PETERLIK, I., AND COTIN, S. 2012. SOFA: A Multi-Model Framework for Interactive Physical Simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Springer. `http://www.sofa-framework.org`.

GALOPPO, N., OTADUY, M. A., MECKLENBURG, P., GROSS, M., AND LIN, M. C. 2006. Fast simulation of deformable models in contact using dynamic deformation textures. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '06, 73–82.

GILLES, B., BOUSQUET, G., FAURE, F., AND PAI, D. 2011. Frame-based Elastic Models. In *ACM Transactions on Graphics*, vol. 30.

Scene	Octopus (Fig.1)			Walrus (Fig.10)			Tourniquet (Fig.9)	
DOFs	10 affine	6878 points	10 affine + 48 points	4566 points	2 rigids + 16 modes	2 rigids + 16 modes + 111 points	7500 points	5 affine + 510 points
Integration Points	10	37912	154	26392	50	390	36450	2440
Frame Rate (Hz)	60	4	35 (pre-factorized) 25 (full solve)	1	100	15 (pre-factorized) 10 (full solve)	<0.1	2 (pre-factorized) 2 (full solve)

Table 1: *Performances*

GRINSPUN, E., KRYSL, P., AND SCHRÖDER, P. 2002. Charms: a simple framework for adaptive simulation. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 21.

HAHN, F., MARTIN, S., THOMASZEWSKI, B., SUMNER, R., COROS, S., AND GROSS, M. 2012. Rig-space physics. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 31.

HARMON, D., AND ZORIN, D. 2013. Subspace integration with local deformations. *ACM Trans. Graph. 32*, 4 (July), 107:1–107:10.

KIM, T., AND JAMES, D. L. 2012. Physics-based character skinning using multidomain subspace deformations. *Visualization and Computer Graphics, IEEE Transactions on 18*, 8, 1228–1240.

MCADAMS, A., ZHU, Y., SELLE, A., EMPEY, M., TAMSTORF, R., TERAN, J., AND SIFAKIS, E. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph. 30*, 4 (July), 37:1–37:12.

MÜLLER, M., DORSEY, J., MCMILLAN, L., JAGNOW, R., AND CUTLER, B. 2002. Stable real-time deformations. In *ACM Transactions on Graphics (Proc. SIGGRAPH)/Eurographics Symposium on Computer Animation*.

NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph. 31*, 6 (Nov.), 152:1–152:10.

NARAIN, R., PFAFF, T., AND O'BRIEN, J. F. 2013. Folding and crumpling adaptive sheets. *ACM Trans. Graph. 32*, 4 (July), 51:1–51:8.

RÉMILLARD, O., AND KRY, P. G. 2013. Embedded thin shells for wrinkle simulation. *ACM Trans. Graph. 32*, 4 (July), 50:1–50:8.

ROHMER, D., POPA, T., CANI, M.-P., HAHMANN, S., AND SHEFFER, A. 2010. Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles. In *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*.

SEILER, M., AND HARDERS, J. S. . M. 2012. Enriching coarse interactive elastic objects with high-resolution data-driven deformations. In *ACM Transactions on Graphics (Proc. SIGGRAPH)/Eurographics Symposium on Computer Animation*.

SIFAKIS, E., SHINAR, T., IRVING, G., AND FEDKIW, R. 2007. Hybrid simulation of deformable solids. In *ACM Transactions on Graphics (Proc. SIGGRAPH)/Eurographics Symposium on Computer Animation*.

SZABÓ, B., DÜSTER, A., AND RANK, E. 2004. The p-version of the finite element method. *Encyclopedia of computational mechanics*.

TERZOPOULOS, D., AND METAXAS, D. 1990. Dynamic 3d models with local and global deformations: deformable superquadrics. In *Computer Vision, 1990. Proceedings, Third International Conference on*, 606–615.

TERZOPOULOS, D., AND WITKIN, A. 1988. Physically based models with rigid and deformable components. *Computer Graphics and Applications, IEEE 8*, 6, 41–51.

TOURNIER, M., NESME, M., FAURE, F., AND GILLES, B. 2014. Velocity-based Adaptivity of Deformable Models. *Computers and Graphics 45* (Dec.), 75 – 85.

TOURNIER, M., NESME, M., GILLES, B., AND FAURE, F. 2015. Stable Constrained Dynamics. *ACM transactions on Graphics, Proceedings of ACM SIGGRAPH*.

TWIGG, C. D., AND KAČIĆ-ALESIĆ, Z. 2010. Point cloud glue: Constraining simulations using the procrustes transform. In *ACM Transactions on Graphics (Proc. SIGGRAPH)/Eurographics Symposium on Computer Animation*.

WANG, H., HECHT, F., RAMAMOORTHI, R., AND O'BRIEN, J. 2010. Example-based wrinkle synthesis for clothing animation. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

WICKE, M., RITCHIE, D., KLINGNER, B. M., BURKE, S., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph. 29*, 4 (July), 49:1–49:11.

WOJTAN, C., AND TURK, G. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. Graph. 27*, 3 (Aug.), 47:1–47:8.

WOJTAN, C., MÜLLER-FISCHER, M., AND BROCHU, T. 2011. Liquid simulation with mesh-based surface tracking. In *ACM SIGGRAPH 2011 Courses*, ACM, New York, NY, USA, SIGGRAPH '11, 8:1–8:84.

ZURDO, J. S., BRITO, J. P., AND OTADUY, M. A. 2013. Animating wrinkles by example on non-skinned cloth. *IEEE Transactions on Visualization and Computer Graphics 19*, 1.

A Off-diagonal blocks for N-levels

In the expression of an upper off-diagonal block $\mathbf{M}_{ij}$, $i < j$, we split $\bar{\mathbf{P}}_{j-1}$:

$$\begin{aligned}\mathbf{M}_{ij} &= \mathbf{J}_i^T \bar{\mathbf{P}}_{i-1}^T \mathbf{M}_e \bar{\mathbf{P}}_{j-1} \mathbf{J}_j \\ &= \mathbf{J}_i^T \bar{\mathbf{P}}_{i-1}^T \mathbf{M}_e \bar{\mathbf{P}}_{i-1} (\mathbf{I} - \mathbf{P}_i) \left(\prod_{k=i+1}^{j-1} (\mathbf{I} - \mathbf{P}_k) \right) \mathbf{J}_j\end{aligned}$$

The first part of this formula is null, due to the properties of our filters:

$$\begin{aligned}&\mathbf{J}_i^T \bar{\mathbf{P}}_{i-1}^T \mathbf{M}_e \bar{\mathbf{P}}_{i-1} (\mathbf{I} - \mathbf{P}_i) = \\ &\mathbf{J}_i^T \bar{\mathbf{P}}_{i-1}^T \mathbf{M}_e \bar{\mathbf{P}}_{i-1} - \underbrace{\mathbf{J}_i^T \bar{\mathbf{P}}_{i-1}^T \mathbf{M}_e \bar{\mathbf{P}}_{i-1} \bar{\mathbf{P}}_{i-1} \mathbf{J}_i}_{\mathbf{M}_i} \mathbf{M}_i^{-1} \mathbf{J}_i^T \bar{\mathbf{P}}_{i-1}^T \mathbf{M}_e \bar{\mathbf{P}}_{i-1} \\ &= \mathbf{0}\end{aligned}$$

The property $\bar{\mathbf{P}}_{i-1} \bar{\mathbf{P}}_{i-1} = \bar{\mathbf{P}}_{i-1}$ holds since $\bar{\mathbf{P}}_{i-1}$ is a projection.