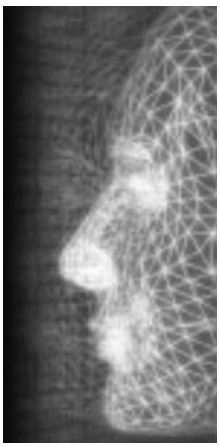
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2010; **21**: 151–159
Published online 29 May 2010 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/cav.373

Real-time hybrid solid simulation: spectral unification of deformable and rigid materials



*By Yin Yang, Guodong Rong, Luis Torres and Xiaohu Guo**

A novel framework is proposed in this paper to simulate hybrid solids with deformable and rigid materials in real-time. Both types of materials are uniformly integrated into one spectral simulator. Based on the modal warping technique, we employ a new constraint strategy which eliminates the accumulation of approximation errors at the boundary interfaces, thus naturally gluing different materials. We also utilize the GPU to accelerate the run-time computation when updating the geometry of the hybrid solid—the most expensive step in this framework. This work provides a general-purpose solution of simulating hybrid objects in real-time, even for large-scale models. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: GPU; hybrid simulation; modal analysis; physically-based animation

Introduction

In physically-based animation, objects are expected to efficiently exhibit realistic motions. Different types of objects typically require particularly designed simulating strategies. The diversity of these techniques which nicely resolve each individual physical phenomenon raises new challenges if we want to have a mixed scenario that either various simulators are to be incorporated[1–3] or if the object under simulation itself is a compound of different physical features.[4–7] The challenge originates from the intrinsic discrepancies among various physical solvers with specialized techniques. The difficulties go further for the hybrid body simulation because of the ambiguity of physical properties at the connecting regions of different materials which calls for a sophisticated constraint strategy.

Inspired by such requirements and challenges we propose a new spectral framework to simulate hybrid solids consisting of both deformable and rigid materials. In this framework, heterogenous materials correspond to sub dynamic systems that evolve in a fully coupled manner in the spectral domain. The spectral reduction of *degrees of freedom* (DoF), as the biggest benefit of spectral simulation technique (also well known as *modal analysis*) results in a significant speed-up over its spatial counterpart.

In addition, the uniform simulation strategy at all the subsystems allows us to construct a unified spectral simulator for the whole hybrid solid. We assemble the global spectral displacement and rotation matrices, and use these matrices to convert spectral displacement to spatial displacement. Because every three rows in these global matrices correspond to a node on the hybrid solid, these global matrices based operations can be naturally parallelized with *graphics processing units* (GPU). Our experiments (Table 1) show that the utilization of the GPU contributes about an extra threefold improvement in terms of time performance. Meanwhile, the symmetric and consistent formulation smooths the whole theory derivation and effectively facilitates the programming implementation.

We extend the *modal warping* method[8] to make the simulation robust to rotation. This technique provides an elegant approach to approximate nodal rotation accompanied with deformation using linear-strain-tensor-based computation. This rotation is also used in our fortified constraint formulation ensuring correct interactivity among subsystems. System degeneration induced by the reinforced boundary condition is alleviated by limiting the number of boundary constraints without introducing much boundary inconsistency (Figure 3). As a result, real-time simulation of hybrid solids with deformable

*Correspondence to: X. Guo, Assistant Professor, Department of Computer Science The University of Texas at Dallas. E-mail: xguo@utdallas.edu

Model	#Tetra.	#Modes	Offline computation (seconds)			Online computation (FPS)	
			Build M, K, C	Solve eigenproblem	Initialize linear system	CPU simulation	GPU simulation
Bar1	323	100	0.008	1.547	0.006	269.1	845.2
Bar2	5372	200	0.419	4.617	0.026	31.9	89.2
Gargoyle	10 000	50	1.547	5.101	0.022	20.3	64.9
Armadillo	13 852	100	2.931	6.823	0.035	14.3	43.6
Dragon	32 959	50	12.209	8.781	0.05	10.3	35.6

Table 1. Time performance.

Figure 1. The hybrid Armadillo model with two rigid bones inside the right leg.

and rigid materials is made possible by utilizing only a small number of modal bases. Figure 1 shows the result of bending one leg of the Armadillo model with 13 852 tetrahedra and 100 spectral bases, where the hybrid leg has two rigid "bone" regions inside.

Related Work

In deformable model, the linear strain tensor is commonly used for the sake of computational efficiency. However, it cannot properly handle rotational deformations. Some researches[9,10] treated the deformations as linear elasticity and rigid motion separately. However, the globally extracted rotation is still not precise enough for large rotations. The *domain decomposition method* (DDM)[11] was proposed as a solution to this problem where the rotation is computed at each subdomain. Alternatively, Müller *et al.*[12] invited *stiffness warping* where the stiffness is warped according to an estimated nodal rotation.

Modal analysis, on the other hand, gives a new perspective to deformation. Pentland and Williams[13] first proposed a basic solution using modal analysis to simulate a deformable body. It was pointed out that the modes associated with higher resonance have less effect on the shape of the object. This property is utilized in almost all the research related to modal analysis because by discarding the high frequency components, the original system is reduced to a new subspace of much smaller size. Choi and Ko[8] transplanted the *warping* idea into spectral space. In their work, rotation is calculated as the curl of the displacement field expressed using spectral displacement. Hence, similar to stiffness warping, linear-strain-tensor-based computation is able to approximate rotational deformation as well in spectral space. This technique is further extended to meshless simulation[14] and thin shell simulation.[15] Besides algorithms, the rapid development of graphics hardware also provides new research space on this topic.[16,17] With more powerful programable graphics hardware, the time performance can be improved even further.

Mixed simulation of multiple physical simulators is more involved, as the interaction and coordination among the different simulators generate more complexity. We roughly classify this type of research into three categories. References [2–4] are several top paradigms of the first category where multiple physically heterogenous scenarios are simulated simultaneously. Coupling of these objects as well as the simulators behind them is the main objective of this type of research. As a result, collision or contact detection is a must because the simulators are not truly coupled until the objects are interacting with each other. Another category of research featured in References [5,18,19], on the other hand, focuses on only one object that more or less demonstrates mixed physical characteristics. Certain hierarchical auxiliary structures in addition to the original mesh such as coarse meshes or grids are commonly employed to facilitate simulation. Although computation based on lower resolution meshes is much faster, this advantage is some-

DOI: 10.1002/cav

how overshadowed by the additional expense of converting, embedding or mapping between multiple resolutions. The last category[6,7,10] puts considerable efforts on physically solving the internal connections among different objects or components. Unfortunately, the rotation at the interface between heterogeneous materials is not clearly addressed: Galoppo *et al.*[10] used the uniform rigid body rotation during simulation. This approach, to our understanding, could lead to some unnatural results when there is large rotational deformation and in References [6,7], certain constraint based formulation were introduced without explicitly comments on rotation handling while we explicitly handle such rotation based on finite strain theory. In the following sections, the detailed build-up of our framework is described.

Spectral Simulation of Deformable and Rigid Regions

We use the subscript $_d$ to denote the values for deformable regions. The Euler–Lagrange equation of a deformable body in $\mathbb{R}^3$ discretized using the finite element method (FEM) is:

$$\mathbf{M}_d\ddot{\mathbf{u}}_d + \mathbf{C}_d\dot{\mathbf{u}}_d + \mathbf{K}_d\mathbf{u}_d = \mathbf{f}_d \tag{1}$$

where $\mathbf{u}_d$ is a time-dependent vector representing the spatial displacement of the object; $\mathbf{K}_d$ and $\mathbf{M}_d$ are the *stiffness matrix* and *mass matrix*, respectively; $\mathbf{f}_d$ is the external force; $\mathbf{C}_d$ is the *Rayleigh damping matrix* as commonly adopted, which is the linear combination of $\mathbf{K}_d$ and $\mathbf{M}_d$, i.e., $\mathbf{C}_d = \xi\mathbf{M}_d + \zeta\mathbf{K}_d$, where ξ and ζ are two weighting constants.

The generalized eigenproblem defined as:

$$\mathbf{K}_d\mathbf{\Phi}_d = \mathbf{M}_d\mathbf{\Phi}_d\mathbf{\Lambda}_d \tag{2}$$

is solved, where $\mathbf{\Phi}_d$ is called the *modal displacement matrix* whose columns are the eigenvectors; $\mathbf{\Lambda}_d$ is a diagonal matrix and the diagonal elements are the corresponding eigenvalues. With the computed eigenvectors, the spatial displacement $\mathbf{u}_d$ can be expressed as:

$$\mathbf{u}_d = \mathbf{\Phi}_d\mathbf{q}_d \tag{3}$$

where $\mathbf{q}_d$ is called the *spectral displacement*. All the spatial unknowns are to be re-expressed in the spectral version of Equation (1):

$$\hat{\mathbf{M}}_d\ddot{\mathbf{q}}_d + \hat{\mathbf{C}}_d\dot{\mathbf{q}}_d + \hat{\mathbf{K}}_d\mathbf{q}_d = \hat{\mathbf{f}}_d \tag{4}$$

where $\hat{\mathbf{f}}_d = \mathbf{\Phi}_d^\top\mathbf{f}_d$ is the *spectral force*. A convenient side-effect is that $\hat{\mathbf{M}}_d = \mathbf{I}$ ($\mathbf{I}$ is the identity matrix), $\hat{\mathbf{K}}_d = \mathbf{\Lambda}_d$ and $\hat{\mathbf{C}}_d = (\xi\mathbf{I} + \zeta\mathbf{\Lambda}_d)$ are all diagonalized.

The rotation is handled using the modal warping[8] method. With this technique, the rotation at each node in the deformable region is extracted and tracked by a 3×1 *rotation vector* $\mathbf{w}$, whose direction and magnitude denote the rotation axis and angle, respectively. The rotation vector can be computed as the *curl* of the displacement field. Then we can build a *modal rotation matrix* $\mathbf{\Psi}$ as a counterpart of $\mathbf{\Phi}$ such that:

$$\mathbf{w}_d = \frac{1}{2}(\nabla\times)\mathbf{H}\mathbf{\Phi}_d\mathbf{q}_d = \mathbf{\Psi}_d\mathbf{q}_d \tag{5}$$

Here $\mathbf{H}$ is the shape function of the tetrahedra mesh, $\mathbf{w}_d$ is a $3 \times n$ vector concatenating the rotation at every node. When advancing simulation, we embed a *local coordinate frame* at each node. Now the motion within one time step of the deformable region can be re-expressed as:

$$\hat{\mathbf{M}}_d\ddot{\mathbf{q}}_d + \hat{\mathbf{C}}_d\dot{\mathbf{q}}_d + \hat{\mathbf{K}}_d\mathbf{q}_d = \mathbf{\Phi}_d^\top(\mathbf{R}_d^\top\mathbf{f}_d) \tag{6}$$

where $\mathbf{R}_d$ is the rotation matrix computed from $\mathbf{w}_d$. $\ddot{\mathbf{q}}$, $\dot{\mathbf{q}}$, and $\mathbf{q}$ correspond to the *local* quantities as mentioned. The final spatial displacement used for updating the geometry of the deformable region can be computed through:

$$\mathbf{u}_d = \tilde{\mathbf{R}}_d\mathbf{\Phi}_d\mathbf{q}_d \tag{7}$$

where $\tilde{\mathbf{R}}_d$ is the accumulated rotation at each node from the beginning of simulation to the current time step. The detailed derivation can be found in Reference [8].

Clearly, fewer modes used bring more deformation details loss. However, in our experiment, it is typically hard to tell by sight such difference and some previous researches[8,13] also justify the high visual plausibility of using spectral simulation.

Because both $\mathbf{M}$ and $\mathbf{K}$ are positive semi-definite matrices, all the eigenvalues from Equation (2) are greater than or equal to zero. Interestingly, there are always six eigenvalues equal to zero. This implies that these six spectral modes represent a dynamics with a zero spectral stiffness matrix which, naturally, results in a zero-strain over the body. Such dynamics is precisely the rigid body motion and the number of modes equals to the number

DOI: 10.1002/cav

of DoF needed in the spatial case, i.e., three for the position and three for rotation. Based on the above analysis, the formulation for simulating the rigid region can be directly incorporated as:

$$\hat{\mathbf{M}}_r\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}_r\dot{\mathbf{q}}_r + \hat{\mathbf{K}}_r\mathbf{q}_r = \mathbf{\Phi}_r^\top(\mathbf{R}_r^\top\mathbf{f}_r) \tag{8}$$

Similarly, we use the subscript $_r$ to represent the values of rigid regions. The number of spectral bases employed is the only difference between deformable and rigid regions.

Without loss of generality, we suppose the hybrid material consists of one deformable region and one rigid region. We can then unify all subsystems into one as:

$$\begin{cases} \hat{\mathbf{M}}_d\ddot{\mathbf{q}}_d + \hat{\mathbf{C}}_d\dot{\mathbf{q}}_d + \hat{\mathbf{K}}_d\mathbf{q}_d = \mathbf{\Phi}_d^\top(\mathbf{R}_d^\top\mathbf{f}_d) \\ \hat{\mathbf{M}}_r\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}_r\dot{\mathbf{q}}_r + \hat{\mathbf{K}}_r\mathbf{q}_r = \mathbf{\Phi}_r^\top(\mathbf{R}_r^\top\mathbf{f}_r) \end{cases} \tag{9}$$

where $\mathbf{f}_d$ and $\mathbf{f}_r$ include not only the external forces applied to each subsystem but also the internal forces between the subsystems.

Boundary Handling

The explicit computation of these internal forces is not trivial. However, these forces can be naturally expressed in the form of constraints which make the duplicated boundary nodes overlap at the interface: $\mathbf{u}_d^b = \mathbf{u}_r^b$, where $\mathbf{u}_d^b$ and $\mathbf{u}_r^b$ stand for the nodal displacements from the deformable region and the rigid region, respectively. One must note that regions of the same type sharing one or more triangle faces are counted as one. Thus the concept of boundary is only the interface between a pair of deformable and rigid regions. Using Equation (7) we have:

$$\mathbf{E}_d^b\tilde{\mathbf{R}}_d\mathbf{\Phi}_d\mathbf{q}_d = \mathbf{E}_r^b\tilde{\mathbf{R}}_r\mathbf{\Phi}_r\mathbf{q}_r \tag{10}$$

where $\mathbf{E}_d^b$ and $\mathbf{E}_r^b$ are the *mapping matrices*. The displacement at the boundary nodes are picked out by premultiplying $\mathbf{E}^b$ to the global displacement $\mathbf{u}$ such that: $\mathbf{u}^b = \mathbf{E}^b\mathbf{u}$. The mapping matrices can be easily built once the partition of the deformable region and the rigid region is finished.

Equation (10), however, does not yield expected results. Our experiments show that the strain accumulates at the boundary and suddenly releases periodically, which causes oscillation at the boundary and the system is highly unstable under rotation. The reason behind this problem is the accumulated rotation term, $\tilde{\mathbf{R}}$ in Equation 10. According to finite strain theory, the deformation can be considered as a combination of a linear elastic deformation and an infinitesimal rotation. Though the first term is computed accurately with the linear tensor, the rotation term is approximated using modal warping. The errors generated will accumulate and make the system unstable. Accordingly, we propose a new constraint formulation following the same idea of finite strain theory that decomposes the boundary constraint into two types of subconstraints of linear elastic and rotation, respectively. In the other words, the change of both linear deformation and rotation within one time step are forced to be equal from two neighboring regions. The modal displacement matrix and modal rotation matrix facilitate the extraction of the linear deformation and rotation symmetrically:

$$\begin{cases} \mathbf{E}_d^b\mathbf{\Phi}_d\mathbf{q}_d = \mathbf{E}_r^b\mathbf{\Phi}_r\mathbf{q}_r \\ \mathbf{E}_d^b\mathbf{\Psi}_d\mathbf{q}_d = \mathbf{E}_r^b\mathbf{\Psi}_r\mathbf{q}_r \end{cases} \tag{11}$$

If the warping technique is to combine the linear rotation and deformation together, this constraint formulation is to inversely separate these two components. With the explicitly defined linear rotation and deformation, the approximation error does not accumulate or cause any stability issues.

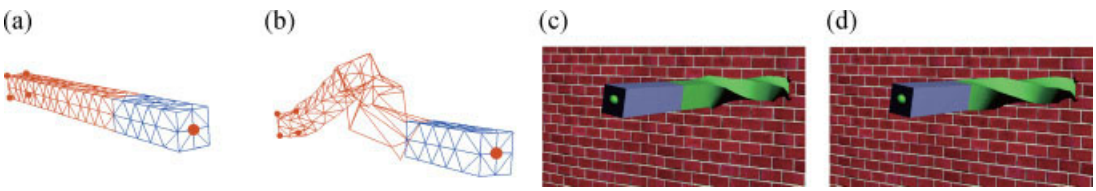


Figure 2. A hybrid bar model consisting of 323 tetrahedra is twisted: (a) The rest shape, (b) Equation (10) is used as boundary constraint, (c) all eight boundary nodes are constrained (over-constrained), and (d) Only two nodes on the boundary are constrained (over-constraint resolved).

DOI: 10.1002/cav

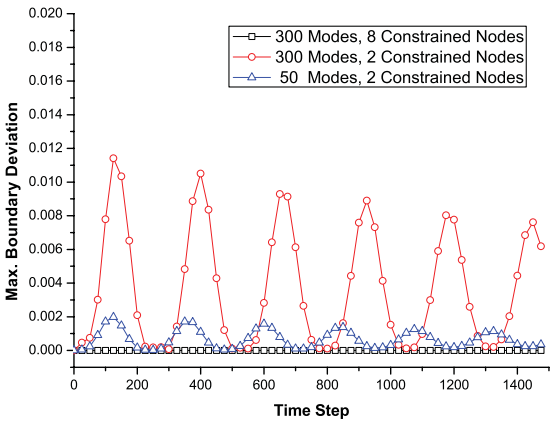


Figure 3. Boundary deviation comparisons with different number of modes and boundary constraint nodes.

For every constrained boundary node, the number of vanished DoF raises from 3 (using Equation (10)) to 6 (using Equation (11)), indicating that twice as many constraints are induced. Importing more constraints into the system, however, brings overhead: the increased number of eliminated DoF due to boundary constraints could turn out to degenerate the system, i.e., there may not be enough available DoF to allow the body to deform properly. This causes the system to be *over-constrained* as shown in Figure 2(c). One solution could be to simply increase the number of spectral bases used. However, this will weaken the major benefit of spectral simulation. Alternatively, we resolve this problem by capping the number of constrained nodes at the boundary as shown in Figure 2(d) where we only constrain two boundary nodes and the bar behaves naturally. The reader may wonder whether this partially constrained boundary still bonds two neighboring regions with high quality. The answer is yes, because our size-reduced spectral simulator implicitly neutralizes the demand of the number of boundary constraints. To be more specific, the deformable regions in the hybrid solid are actually becoming 'stiffer' due to the absence of high frequency vibrations. As a side-effect, the stiffened boundary requires fewer boundary constraints. An extreme example may explain this more clearly: if the deformable region degenerates to a subsystem of only six DoF, it actually becomes a purely rigid one; as a result, a single boundary node will suffice to connect the this subsystem with its rigid neighbor. Of course, the boundary nodes that are not constrained can still experience slight deviations. Figure 3 quantifies such a situation. The curves are generated from a bar model with 323 tetrahedra (same model as in Figure 2). One end of the bar is fixed and the other end drops freely under gravity. The deviation is the boundary nodal displacement difference between the deformable region and the rigid region. This quantity evaluates how well two materials are physically glued by the boundary constraint. In order to give the reader a better understanding of the values of deviations in the figure, the bar model is normalized into a $1 \times 1 \times 1$ cube. As shown in the figure, when all the boundary nodes are constrained (square dot curve) the deviation is very close to its analytical value, 0. However if the constrained boundary nodes are reduced to two, a system with 50 modes (triangle dot curve) satisfies the boundary condition much better than a system with 300 modes (round dot curve) does. As we always want to use smaller number of spectral bases to reduce the size of the system, 50 modes are always preferred over 300 modes. Such tiny errors at the boundary ($< 0.2\%$ of model dimension) are negligible during simulation and do not cause force feedback (the interactivity between subsystems are affected only by constrained boundary nodes) and thus do not affect the system's stability.

The selection of constrained boundary nodes is not arbitrary, as we always want the constrained nodes to be evenly scattered over the boundary. In other words, the set of chosen constrained nodes C is a subset of boundary nodes B such that the *distance* between C and $B - C$ is minimized, assuming each edge on the boundary is equally weighted. We employ a fast, simple algorithm to select C from B as follows:

- An undirected connected graph $G(V, E)$ is constructed from the topology of the boundary. Each node on the boundary corresponds to a vertex in V and every edge connecting two nodes on the boundary corresponds to an undirected edge, $e \in E$.
- An *adjacency matrix* $\mathcal{D}^0$ from G is computed such that $\forall d_{i,j} \in \mathcal{D}^0$, $d_{i,j} = \begin{cases} 1, & \text{if } e(i,j) \in E \\ 0, & \text{otherwise.} \end{cases}$ and $\mathcal{D}^{\frac{|B|}{2}}$ is computed using the *Floyd–Warshall algorithm*.†
- The first node is chosen as the one with the largest number of one-ring neighbors (cover the largest area). The next candidate is $b \in B - C$ such that $\forall c, c \in C$, $min\{Distance(c, b)\}$ is maximized, where $Distance(c, b)$ denotes the number of edges on the shortest patch connecting c and b in G. This value can be found in $\mathcal{D}^{\frac{|B|}{2}}$.

†The degree of the adjacency matrix is $\frac{|B|}{2}$ because G is a *biconnected* graph: the removal of any vertex on the graph does not affect the connectivity of the resulting graph. $\frac{|B|}{2}$ guarantees that the length of *all pair shortest path* is evaluated correctly.

DOI: 10.1002/cav

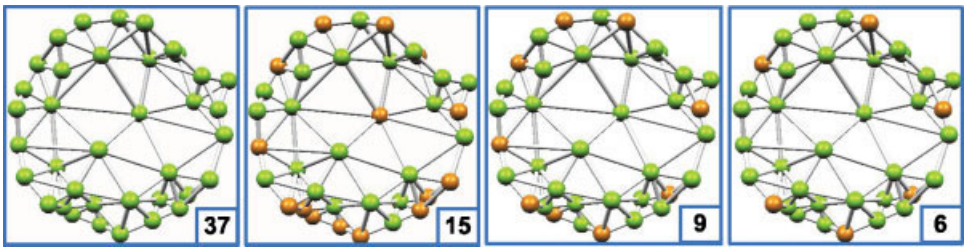


Figure 4. Results of boundary node selection. Selected nodes are shown in orange.

Figure 4 shows the resulting selection of 15th, 9th, and 6th nodes out of the 37 boundary nodes.

Time Integration and Constraint Manipulation

A coupled ODE must be solved at each time step in every subsystem. The *Newmark family* is the most widely used family of direct methods for solving systems like $\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}$. Here we use an implicit family member, *average acceleration method* to linear the ODE as $\mathbf{A}\ddot{\mathbf{u}} = \mathbf{b}$, where $\mathbf{A} = \mathbf{M} + \gamma h\mathbf{C} + \beta h^2\mathbf{K}$, $\mathbf{b} = \mathbf{f}^{n+1} - \mathbf{C}\tilde{\dot{\mathbf{u}}}^{n+1} - \mathbf{K}\tilde{\mathbf{u}}^{n+1}$, $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$. $\tilde{\mathbf{u}}^{n+1}$ and $\tilde{\dot{\mathbf{u}}}^{n+1}$ are two *predictors* defined as:

$$\begin{cases} \tilde{\mathbf{u}}^{n+1} = \mathbf{u}^n + h\dot{\mathbf{u}}^n + \frac{h^2}{2}(1-2\beta)\ddot{\mathbf{u}}^n, \\ \tilde{\dot{\mathbf{u}}}^{n+1} = \dot{\mathbf{u}}^n + (1-\gamma)h\ddot{\mathbf{u}}^n \end{cases} \tag{12}$$

The unknown displacements and velocities can be computed through:

$$\begin{cases} \mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} + \beta h^2\ddot{\mathbf{u}}^{n+1}, \\ \dot{\mathbf{u}}^{n+1} = \tilde{\dot{\mathbf{u}}}^{n+1} + \gamma h\ddot{\mathbf{u}}^{n+1} \end{cases} \tag{13}$$

In our implementation, the variables are the spectral counterparts, i.e. q. Constraints expressed as linear equations are integrated using the *Lagrange Multiplier Method*. In addition to the boundary constraint (Equation (11)), other users' manipulation constraints including either position or orientation constraints are necessary as well. These constraints can be formulated in a symmetric fashion. Then we can build a generalized constraint matrix demoted by $\mathbf{J}$ such that:

$$\mathbf{J}\begin{bmatrix}\mathbf{q}_d \\ \mathbf{q}_r\end{bmatrix} = \begin{bmatrix}\mathbf{0} \\ \mathbf{c}\end{bmatrix} \tag{14}$$

where

$$\mathbf{J} = \begin{bmatrix} \mathbf{E}_d^b\boldsymbol{\Phi}_d & -\mathbf{E}_r^b\boldsymbol{\Phi}_r \\ \mathbf{E}_d^b\boldsymbol{\Psi}_d & -\mathbf{E}_r^b\boldsymbol{\Psi}_r \\ \mathbf{E}_d^p\tilde{\mathbf{R}}_d\boldsymbol{\Phi}_d & \mathbf{0} \\ \mathbf{E}_d^o\tilde{\mathbf{R}}_d\boldsymbol{\Psi}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_r^p\tilde{\mathbf{R}}_r\boldsymbol{\Phi}_r \\ \mathbf{0} & \mathbf{E}_r^o\tilde{\mathbf{R}}_r\boldsymbol{\Psi}_r \end{bmatrix} \tag{15}$$

$\mathbf{E}^p$ and $\mathbf{E}^o$ play roles similar to $\mathbf{E}^b$, i.e., picking out the corresponding constrained nodal values of each subsystem. Vector $\mathbf{c}$ is assembled by the desired constraint values. As in Reference [20], the damped second-order constraint form is adopted:

$$\begin{bmatrix}\begin{bmatrix}\mathbf{A}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_r\end{bmatrix} & \tilde{\mathbf{J}}^\top \\ \tilde{\mathbf{J}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{q}}_d \\ \ddot{\mathbf{q}}_r \\ \beta h^2\lambda\end{bmatrix} = \begin{bmatrix}\mathbf{b}_d \\ \mathbf{b}_r \\ \begin{bmatrix}\mathbf{0} \\ \begin{bmatrix}\tilde{\mathbf{R}}_d^\top & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{R}}_r^\top\end{bmatrix}\mathbf{c}\end{bmatrix} - 2p_1\dot{\mathbf{C}} - p_2^2\mathbf{C}\end{bmatrix} \tag{16}$$

where λ denotes the *Lagrange Multipliers*, and p_1 and p_2 are stabilization factors. The rotation terms are moved to the right side of the equation so the matrix on the left-hand side can be pre-computed and $\mathbf{J}$ turns to ($\tilde{\mathbf{J}}$). From the computed $\ddot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_r$ we can get $\mathbf{q}_d$ and $\mathbf{q}_r$ as in Equation (13). The spatial displacements for updating geometry can be computed using Equation (7).

DOI: 10.1002/cav

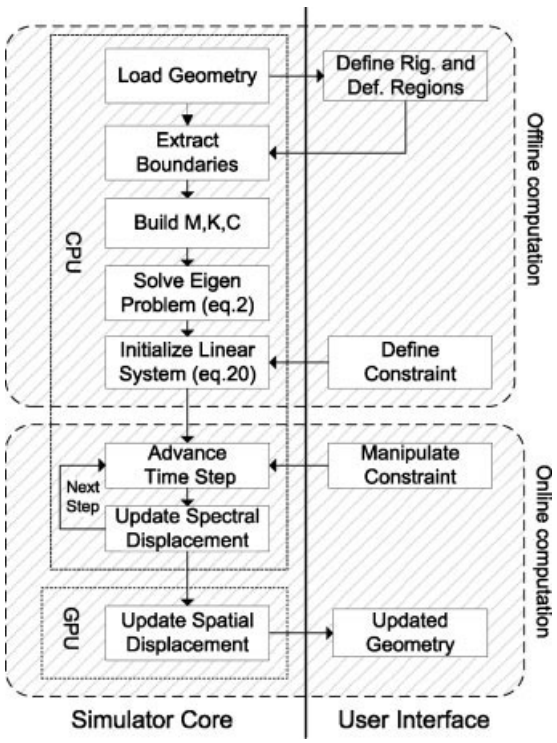


Figure 5. Computational flow chart.

Implementation and Experimental Results

Our framework is implemented using Microsoft Visual C++ 2005 on a Windows XP PC with Intel Core2 Duo 2.93GHz CPU, 2GB DDR2 RAM, and NVIDIA GeForce GTX 280 GPU with 1GB DDR3 VRAM.

Figure 5 is a general flow chart illustrating the computational procedure of this framework. All the computation is partitioned into offline computation and online computation. The spatial geometry update is executed by the GPU. The original hybrid solid is subdivided into several regions as defined by the user. Once the linear system in Equation (16) is resolved, the framework proceeds to online computation.

The most time consuming part in online computation is updating the geometry of the hybrid solid at each time step as in Equation (7). Fortunately, this part can be parallelized with the GPU in our framework. In order to do this, $\mathbf{\Phi}_d$ or $\mathbf{\Phi}_r$ for all the subsystems are assembled into one single global matrix $\bar{\mathbf{\Phi}}$ and $\bar{\mathbf{\Psi}}$ called *global displacement matrix* and *global rotation matrix*, respectively. Similarly, we also build a *global spectral displacement vector* $\bar{\mathbf{q}}$ and *global spectral rotation vector* $\bar{\mathbf{w}}$. Every three consecutive rows correspond to the spatial position or rotation of one node and are updated in parallel with *vertex programs* executing concurrently on multiple nodes.

We have conducted extensive experiments and tests on various 3D tetrahedral meshes. Some of the models are of very large size and not commonly seen in other real-time physically-based animation applications. Detailed benchmarks of each computational stage of our framework are shown in Table 1 where the usage of the GPU makes simulation about three times faster.

Hybrid solids provide more natural solutions to simulate real world objects. In Figure 6, the leg motion of the Armadillo is simulated where two rigid interior regions marked in dark red function as bones. Figure 7 contains several snapshots of our implementation where the user interactively defines the regions' properties and manipulates the models in real-time. Most experiments mentioned can be found in the accompanying video demo which was produced by real-time screen capture of our implementation.

Further Work

This work provides a spectral solution of real-time simulation of hybrid objects. However, currently we have not integrated collision or contact handling into our framework. Though we believe due to the uniformity of this

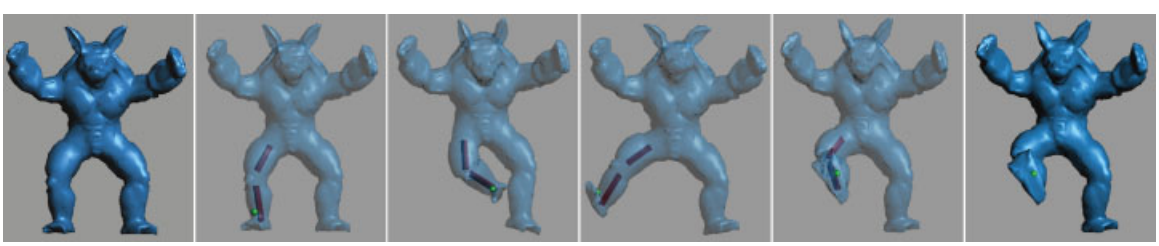


Figure 6. Simulate leg behavior of Armadillo model.

DOI: 10.1002/cav

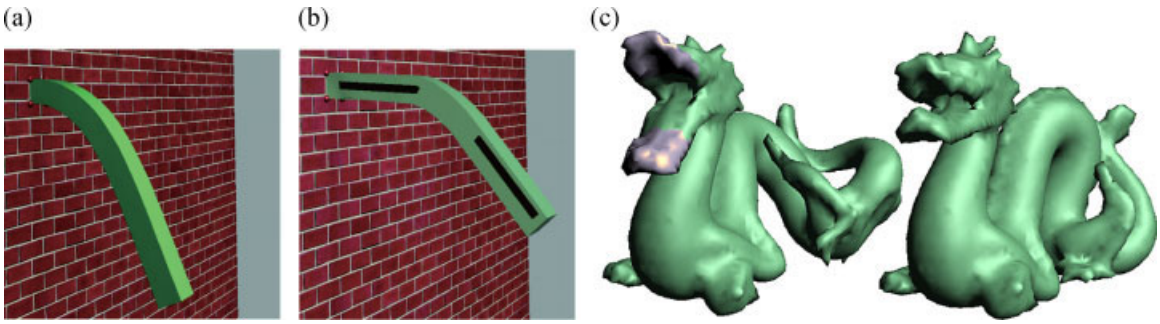


Figure 7. (a) Deformable bar, (b) hybrid bar (5372 tetrahedra), and (c) the dragon model of 32 959 tetrahedra opens its mouth with two rigid jaws (purple).

framework, collision, and contact could be incorporated in a straightforward manner. In addition, the rapid development of graphical hardware raises some more advanced parallel computation languages such as CUDA. As this work is heavily involved with matrices and vectors based computation, these new-emerging programming tools can definitely contribute a faster simulation and benefit the final performance. On the other hand, the rigid region discretized by the 3D tetrahedral mesh, in many situations may not be the best candidate for efficiently describing some natural constraints. One example is the human body, where the rigid subsystem could be more properly expressed using only line-based rigid constraints.[5] In the future, we will investigate different types of rigid subsystems as well as some joint hierarchies models[21] and attempt to integrate them into the spectral hybrid framework to simulate more realistic behaviors of natural creatures in the real world.

ACKNOWLEDGMENTS

Authors thank the anonymous reviewer's diligent efforts and beneficial comments on the paper. This work is supported by the National Science Foundation under Grant No. CCF-0727098.

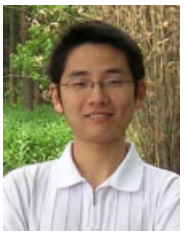
References

1. Baraff D, Witkin A. Partitioned dynamics. *Technical Report CMU-RI-TR-97-33*, Robotics Institute, Carnegie Mellon University, 1997.
2. Tamar S, Craig S, Ronald F. Two-way coupling of rigid and deformable bodies. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2008; 95–103.
3. Robinson-Mosher A, Shinar T, Gretarsson J, Su J, Fedkiw R. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Transactions on Graphics* 2008; **27**(3): 1–9.
4. Sifakis E, Shinar T, Irving G, Fedkiw R. Hybrid simulation of deformable solids. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2007; 81–90.
5. Capell S, Green S, Curless B, Duchamp T, Popović Z. Interactive skeleton-driven dynamic deformations. In *Proceedings of the SIGGRAPH '02*, 2002; 586–593.
6. Johan J, Joris V. Combining deformable and rigid-body mechanics simulation. *The Visual Computer* 2003; **19**: 280–290.
7. Lenoir J, Fonteneau S. Mixing deformable and rigid-body mechanics simulation. In *Proceedings of the Computer Graphics International*, 2004; 327–334.
8. Choi MG, Ko H-S. Modal warping: real-time simulation of large rotational deformation and manipulation. *IEEE Transactions on Visualization and Computer Graphics* 2005; **11**(1): 91–101.
9. Terzopoulos D, Witkin A. Physically based models with rigid and deformable components. *IEEE Computer Graphics and Applications* 1988; **8**(6): 41–51.
10. Galoppo N, Otaduy MA, Mecklenburg P, Gross M, Lin MC. Fast simulation of deformable models in contact using dynamic deformation textures. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006; 73–82.
11. Huang J, Liu X, Bao H, Guo B, Shum H-Y. An efficient large deformation method using domain decomposition. *Computer Graphics* 2006; **30**(6): 927–935.
12. Müller M, Dorsey J, McMillan L, Jagnow R, Cutler B. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002; 49–54.
13. Pentland A, Williams J. Good vibrations: modal dynamics for graphics and animation. *SIGGRAPH Computer Graphics* 1989; **23**(3): 207–214.
14. Guo X, Qin H. Real-time meshless deformation: collision detection and deformable objects. *Computer Animation and Virtual Worlds* 2005; **16**(3–4): 189–200.
15. Choi MG, Woo SY, Ko H-S. Real-time simulation of thin shells. In *Proceedings of Eurographics '07*, 2007; 349–354.
16. James DL, Pai DK. DyRT: dynamic response textures for real time deformation simulation with graphics hardware. In *Proceedings of SIGGRAPH '02*, 2002; 582–585.
17. Raghuvanshi N, Lloyd B, Govindaraju KN, Lin M. Efficient numerical acoustic simulation on graphics processors using

DOI: 10.1002/cav

adaptive rectangular decomposition. In *European Acoustics Association Symposium on Auralization*, 2009.
18. Kharevych L, Mullen P, Owhadi H, Desbrun M. Numerical coarsening of inhomogeneous elastic materials. In *SIGGRAPH '09*, 2009; 1–8.
19. Nesme M, Kry PG, Jeřábková L, Faure F. Preserving topology and elasticity for embedded deformable models. In *SIGGRAPH '09*, 2009; 1–9.
20. Metaxas D, Terzopoulos D. Dynamic deformation of solid primitives with constraints. *SIGGRAPH Computer Graphics* 1992; **26**(2): 309–312.
21. Xu W, Wang J, Yin K, *et al*. Joint-aware manipulation of deformable models. *ACM Transactions on Graphics* 2009; **28**(3): 1–9.

Authors' biographies:



Yin Yang is currently a Ph.D student at Department of Computer Science, University of Texas at Dallas. His research interests include computer graphics and animation, visualization and medical imaging, especially in physically based simulation and animation.

Guodong Rong received his B.Eng. and M.Eng. degrees both in computer science from Shandong University in 2000 and 2003 respectively, and his Ph.D. degree in computer science from National University of Singapore in 2007. He is currently a research scholar at Department of Computer Science, University of Texas at Dallas. His research interests include computer graphics, computational geometry, visualization and image processing, especially on using the GPU to accelerate geometry-related problems.

Luis Torres is a undergraduate student at Department of Computer Science, University of Texas at Dallas. His research focuses on computer graphics.

Xiaohu Guo is an assistant professor of computer science at the University of Texas at Dallas. He received the PhD degree in computer science from the State University of New York at Stony Brook in 2006. His research interests include computer graphics, animation and visualization, with an emphasis on geometric and physics-based modeling. His current researches at UT-Dallas include: spectral geometric analysis, deformable models, centroidal Voronoi tessellation, GPU algorithms, 3D and 4D medical image analysis, etc. For more information, please visit http://www.utdallas.edu/~xguo.

DOI: 10.1002/cav