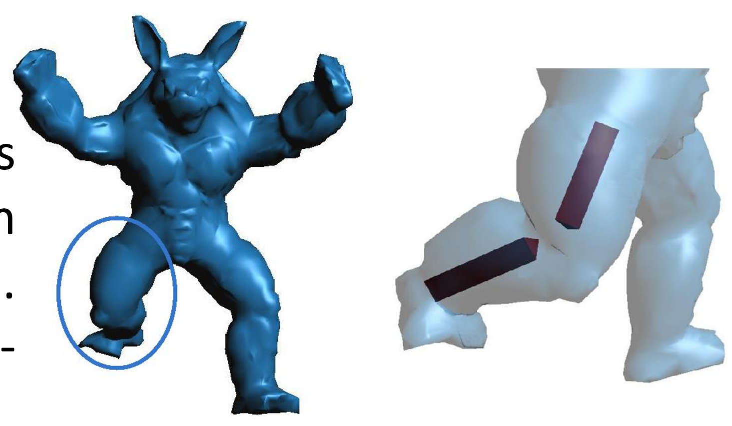
Spectral Simulation of Hybrid Bodies with Deformable and Rigid Materials

Yin Yang Guodong Rong Luis Torres Xiaohu Guo

Abstract

A novel spectral framework is proposed in order to simulate a hybrid body with both deformable and rigid materials in real-time. We extend the modal warping technique from deformable models to rigid bodies and employ a new constraint strategy to eliminate the accumulation of approximation errors at the boundary. The powerful parallel computation ability of the GPU is utilized as well to update the geometry of the hybrid object. This work provides a general-purpose solution to simulating hybrid objects in real-time, even for large-scale models.

Spectral Simulation



The Euler-Lagrange equation is projected onto spectral domain by solving an eigenproblem. After that, the motion is re-represented as the combination of independent vibrations with different frequency. High frequency vibrations are ignored. As a result, the spectral motion equation is of much smaller size than the spectral one.

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

#DoFs: $3 \times n$

Spectral projection + *Modal Warping*

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

#DoFs: $k, k << n$

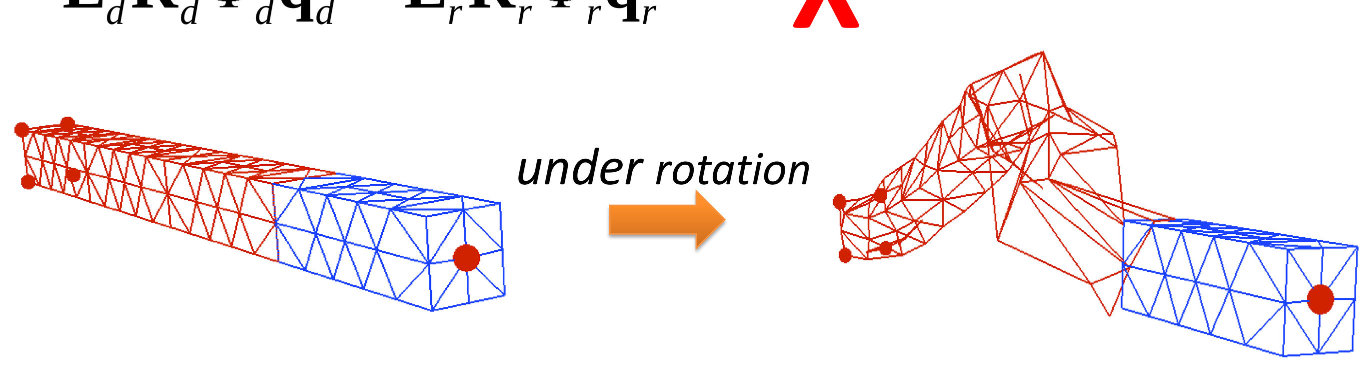
$$\begin{cases} \hat{\mathbf{M}}_d\ddot{\mathbf{q}} + \hat{\mathbf{C}}_d\dot{\mathbf{q}} + \hat{\mathbf{K}}_d\mathbf{q} = \Phi_d^{\mathbf{T}}(\mathbf{R}_d^{\mathbf{T}}\mathbf{f}_d) & \Rightarrow \text{deformable regions} \\ \hat{\mathbf{M}}_r\ddot{\mathbf{q}} + \hat{\mathbf{C}}_r\dot{\mathbf{q}} + \hat{\mathbf{K}}_r\mathbf{q} = \Phi_r^{\mathbf{T}}(\mathbf{R}_r^{\mathbf{T}}\mathbf{f}_r) & \Rightarrow \text{rigid regions} \end{cases}$$

Boundary Handling

The interface from different regions must be overlapped by imposing constraints on the nodes at the interface. Such constraints are called *boundary constraints* : $\mathbf{u}_d^b = \mathbf{u}_r^b$.

- **An intuitive definition**

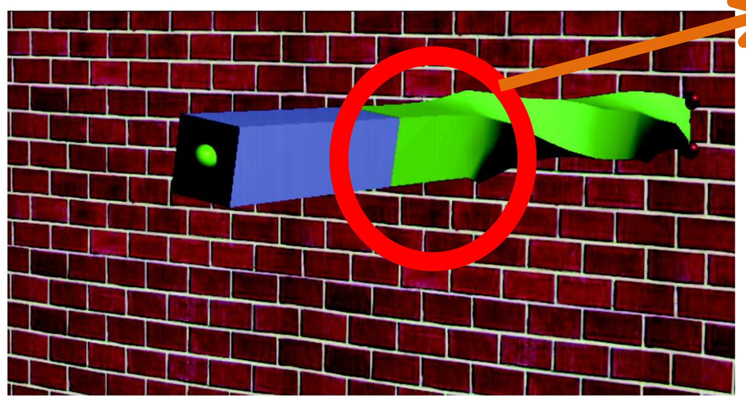
$$\mathbf{E}_d^b\tilde{\mathbf{R}}_d\Phi_d\mathbf{q}_d = \mathbf{E}_r^b\tilde{\mathbf{R}}_r\Phi_r\mathbf{q}_r \quad \text{X}$$



- **A fortified definition**

$$\begin{cases} \mathbf{E}_d^b\Phi_d\mathbf{q}_d = \mathbf{E}_r^b\Phi_r\mathbf{q}_r \\ \mathbf{E}_d^b\Psi_d\mathbf{q}_d = \mathbf{E}_r^b\Psi_r\mathbf{q}_r \end{cases}$$

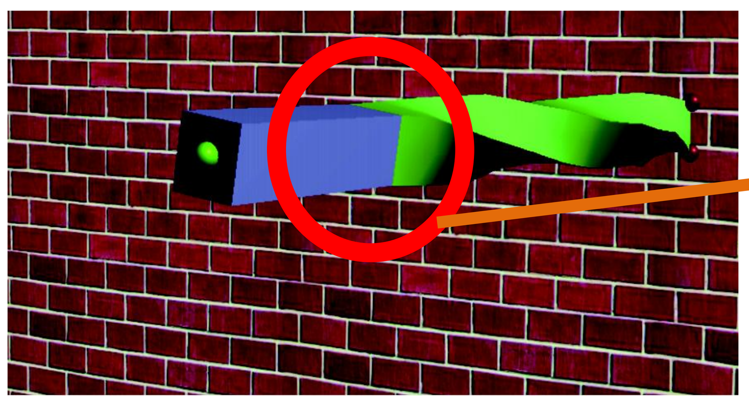
The definition, decompose the boundary constraint into two subconstraints: *1) a constraint on linear deformation 2) a constraint on linear rotation*.



All boundary nodes are constrained

Importing more constraints into the system, however, brings overhead. For every constrained boundary node, the number of vanished DoFs raises from 3 to 6, indicating that twice as many constraints are induced. For our DoF-reduced spectral simulator, 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*.

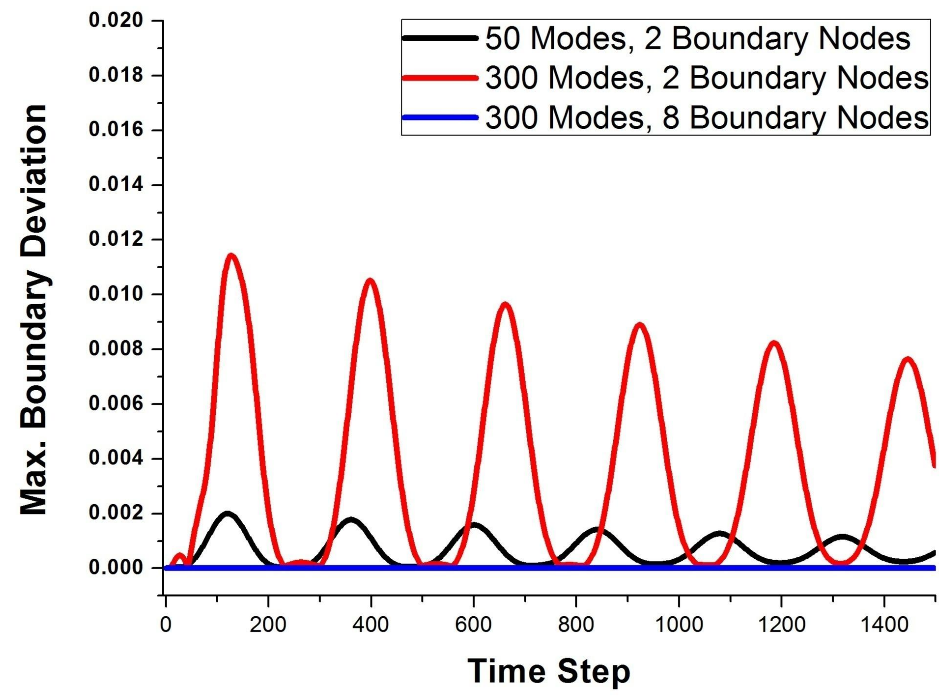
We resolve this problem by capping the number of constrained nodes at the boundary.



The boundary consists of 8 nodes. Only 2 of them are constrained. And the deformable region behaviors naturally

Spectral simulator implicitly neutralizes the demand of the number of boundary constraints. So fewer boundary constraint nodes will not produce much boundary inconsistency.

Boundary deviation is the boundary nodal displacement difference between deformable and rigid regions. If 50 modes are used, the max. deviation is only around 0.2% of mode dimension. This tiny errors is negligible during simulation.



Constraint Manipulation

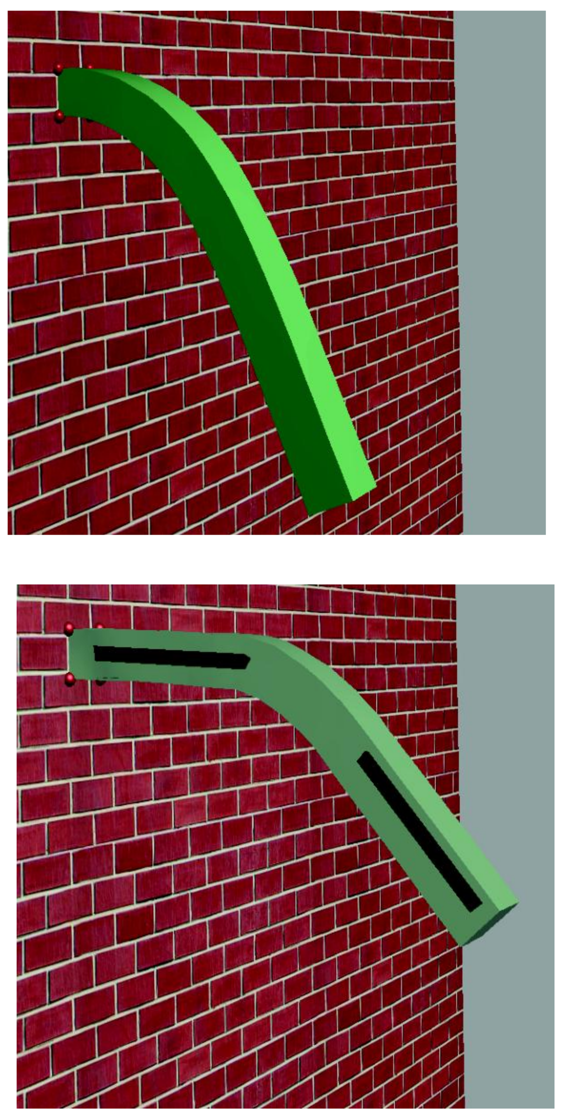
Other users' manipulation constraints including either position or orientation constraints are necessary as well. These constraints can be formulated in a symmetric fashion:

position constraints, **o**rientation constraints

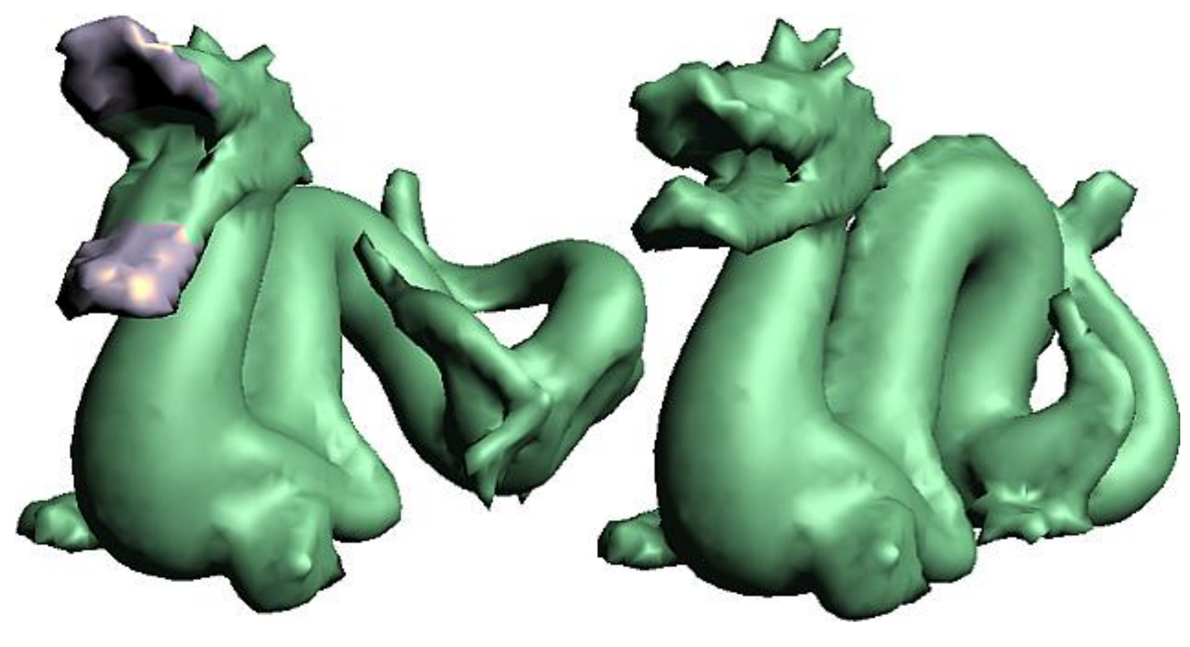
$$\begin{bmatrix} \mathbf{E}_d^p\tilde{\mathbf{R}}_d\Phi_d & 0 \\ \mathbf{E}_d^o\tilde{\mathbf{R}}_d\Psi_d & 0 \\ 0 & \mathbf{E}_r^p\tilde{\mathbf{R}}_r\Phi_r \\ 0 & \mathbf{E}_r^o\tilde{\mathbf{R}}_r\Psi_r \end{bmatrix} \begin{bmatrix} \mathbf{q}_d \\ \mathbf{q}_r \end{bmatrix} = \mathbf{c}$$

Implementation and Results

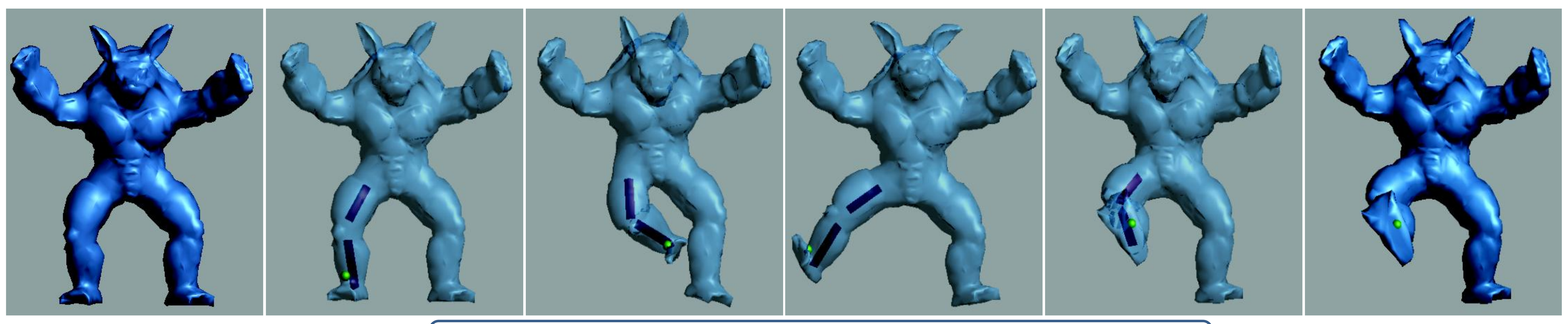
Model	# Tetra	# Mode	CPU Simulation (FPS)	GPU Simulation (FPS)
Bar1	323	100	269.1	845.2
Bar2	5.372	200	31.9	89.2
Gargoyle	10,000	50	20.3	64.9
Armadillo	13,852	100	14.3	43.6
Dragon	32,959	50	10.3	35.6



Deformable bar and hybrid bar (5, 372 tetrahedra)



The dragon model of 32, 959 tetrahedra opens its mouth with two rigid jaws (purple)



Simulate leg behavior of Armadillo model