EUROGRAPHICS 2007 / D. Cohen-Or and P. Slavík
(Guest Editors)

*Volume 26 (2007), Number 3*

# Real-Time Simulation of Thin Shells

Min Gyu Choi[†1], Seung Yong Woo[1], and Hyeong-Seok Ko[2]

[1]Kwangwoon University, Korea
[2]Seoul National University, Korea

**Abstract**

*This paper proposes a real-time simulation technique for thin shells undergoing large deformation. Shells are thin objects such as leaves and papers that can be abstracted as 2D structures. Development of a satisfactory physical model that runs in real-time but produces visually convincing animation of thin shells has been remaining a challenge in computer graphics. Rather than resorting to shell theory which involves the most complex formulations in continuum mechanics, we adopt the energy functions from the discrete shells proposed by Grinspun et al. [GHDS03]. For real-time integration of the governing equation, we develop a modal warping technique for shells. This new simulation framework results from making extensions to the original modal warping technique [CK05] which was developed for the simulation of 3D solids. We report experimental results, which show that the proposed method runs in real-time even for large meshes, and that it can simulate large bending and/or twisting deformations with acceptable realism.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling – Physically Based Modeling; I.6.8 [Simulation and Modeling]: Types of Simulation – Animation

## 1. Introduction

Thin flexible objects such as leaves or papers often appear in computer graphics scenes. Non-zero structural thickness is a factor that influences their dynamic movements. Nevertheless, those objects are often abstracted as two-dimensional (2D) entities, regarding the rest as visual details. We refer thin flexible objects which can be abstracted as 2D entities as *thin shells*. This paper is about physically-based simulation of thin shells.

Simulation of 3D solids has been studied by the graphics community [BJ05, GM97, JF03, MDM*02, TPBF87]. Since thin shells are special cases of 3D solids, one may apply the techniques developed for 3D solids to the simulation of thin shells. Unfortunately, this approach does not produce satisfactory results; Modeling thin shells as 3D elastic solids requires very fine FEM meshes to correctly capture the global bending behavior.

There is a different approach to simulating shells, namely, representing shells as 2D meshes rather than 3D solids. However, accurate modeling and simulation of a thin shell structure with a moderately-sized 2D mesh requires one of the most complex formulations in continuum mechanics, *shell theory* [GZ68].

In this paper, we develop a real-time simulation technique for thin shells undergoing large rotational deformation. Even though the present work represents a shell with a 2D mesh, it does not derive the governing equation from shell theory, but from a simpler model. A pioneering work of this kind in the computer graphics field is *discrete shells* by Grinspun et al [GHDS03]. They formulated dynamics of thin shells by resorting to a discrete model instead of Cosserat models that are normally employed in shell theory. The resulting governing equation of [GHDS03] takes the same form as that of 3D solids, which makes it easy to understand and implement. However, [GHDS03] did not take any particular steps to make time-integration of the equation run in real-time.

Meanwhile, Choi and Ko [CK05] proposed a technique called *modal warping*, which can simulate large rotational deformation of 3D elastic solids in real-time. But [CK05]

† mgchoi@kw.ac.kr, sywoo@cs.kw.ac.kr, ko@graphics.snu.ac.kr

was not intended for thin shells. On the other hand, Hauser et al. [HSO03] proposed a real-time technique that can simulate thin shells. But the technique could not be used for simulating large rotational deformations.

The present work is based on the discrete shells and the modal warping technique; Adopting the dynamic formulation of Grinspun et al. [GHDS03], the present work develops a new simulation framework for thin shells that runs in real-time by extending the modal warping technique [CK05].

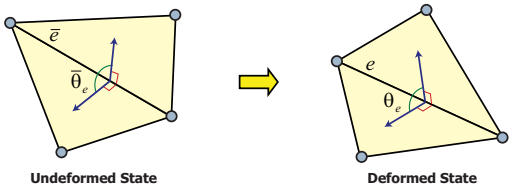


**Figure 1:** *Dihedral angle of an edge in the undeformed and the deformed states.*

## 2. Dynamics of Thin Shell

We use a 2-manifold triangular mesh to represent the shell, and formulate its dynamics with nonlinear membrane and flexure energy functions that measure the differences between the undeformed and the deformed states of the mesh. The energy functions are not a contribution of this paper; All the energy functions that appear in this section are adopted from other people's work. We note that, in fact, the simulation framework we develop in this paper does not depend on the energy functions being used.

### 2.1. Membrane Energy

The membrane energy models the shell resisting on intrinsic deformations, and consists of stretch and shear energies. For the stretch energy, we use the triangle-based function that sums up changes in the area

$$E_A \triangleq \sum_A \frac{\left(\|A\| - \|\bar{A}\|\right)^2}{\|\bar{A}\|}, \tag{1}$$

where $\|A\|$ and $\|\bar{A}\|$ are the areas of the triangle $A$ in the deformed and the undeformed states, respectively.

For the shear energy, we use the edge-based function that sums up changes in the length

$$E_L \triangleq \sum_e \frac{\left(\|e\| - \|\bar{e}\|\right)^2}{\|\bar{e}\|}, \tag{2}$$

where $\|e\|$ and $\|\bar{e}\|$ are the lengths of the edge $e$ in the deformed and the undeformed states, respectively.

### 2.2. Flexural Energy

For measuring the flexural energy of the shell, we use the function proposed by Grinspun et al. [GHDS03],

$$E_B \triangleq \sum_e 3\left(\theta_e - \bar{\theta}_e\right)^2 \frac{\|\bar{e}\|}{\bar{h}_e}, \tag{3}$$

where $\theta_e$ and $\bar{\theta}_e$ are the dihedral angles of the edge $e$ measured in the deformed and the undeformed states, respectively, and $\bar{h}_e$ is the average of the heights of the two triangles sharing the edge $e$ in the undeformed state (See Fig. 1). This energy function was obtained by integrating the squared difference of mean curvature at a point over the piecewise linear mesh of the shell, and then by discretizing the integral (See [GHDS03] for the detailed derivation).

### 2.3. Governing Equation

The total elastic energy of a thin shell is then defined by the sum of the membrane and flexural energies

$$E \triangleq k_A E_A + k_L E_L + k_B E_B, \tag{4}$$

where $k_A$, $k_L$, and $k_B$ are the material constants that represent the stretch, shear, and flexural stiffness, respectively. Differentiating the above energy function with respect to the displacements of the mesh nodes gives the generalized elastic force due to the elastic potential energy, which can be written in the form

$$\frac{\partial E(\mathbf{u})}{\partial \mathbf{u}} = \mathbf{K}(\mathbf{u})\mathbf{u}, \tag{5}$$

where $\mathbf{u}(t)$ is a $3n$-dimensional vector that represents the displacements of the $n$ nodes from their original positions. Then, the governing equation that describes the dynamic movements of a thin shell can be written as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}(\mathbf{u})\mathbf{u} = \mathbf{F}, \tag{6}$$

where $\mathbf{M}$ and $\mathbf{C}$ are the mass and damping matrices, respectively, and $\mathbf{F}(t)$ is a $3n$-dimensional vector that represents the external forces acting on the $n$ nodes. Here, the elastic force term $\mathbf{K}(\mathbf{u})\mathbf{u}$ is nonlinear with respect to $\mathbf{u}$. Therefore real-time integration of Equation (6) is a nontrivial task.

## 3. Modal Warping for Shells

For real-time simulation of thin shells, we adopt the modal warping framework [CK05]. Since the procedure described in [CK05] is for 3D solids, we have to make modifications to make it applicable to shells. A major modification is done to the method to keep track of the orientation of the local coordinate frames associated with the mesh nodes, which is presented in Section 3.2.

### 3.1. Modal Displacements

When there is a small rotational deformation, the generalized elastic force $\mathbf{K}(\mathbf{u})\mathbf{u}$ appearing in Equation (6) can be linearly approximated as $\mathbf{K}\mathbf{u}$ for a constant matrix $\mathbf{K}$. When this

simplification is applicable, we can decouple Equation (6) by solving the generalized eigenvalue problem $\mathbf{K}\Phi = \mathbf{M}\Phi\Lambda$ and finding $\Phi$ and $\Lambda$ such that $\Phi^{\mathrm{T}}\mathbf{M}\Phi = \mathbf{I}$ and $\Phi^{\mathrm{T}}\mathbf{K}\Phi = \Lambda$. Since the columns of $\Phi$ form a basis of the $3n$-dimensional space, $\mathbf{u}$ can be expressed as a linear combination of the columns:

$$\mathbf{u}(t) = \Phi\mathbf{q}(t). \tag{7}$$

Here, $\Phi$ is the *modal displacement matrix*, of which the $i$-th column represents the $i$-th mode shape, and $\mathbf{q}(t)$ is a vector containing the corresponding modal amplitudes as its components. By examining the eigenvalues we can take only the $m$ dominant columns of $\Phi$, significantly reducing the amount of computation. In the following, $\Phi$ denotes the $3n \times m$ submatrix formed by this procedure. Substitution of Equation (7) into Equation (6) followed by a premultiplication of $\Phi^{\mathrm{T}}$ decouples Equation (6) as

$$\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \mathbf{C}_{\mathbf{q}}\dot{\mathbf{q}} + \mathbf{K}_{\mathbf{q}}\mathbf{q} = \Phi^{\mathrm{T}}\mathbf{F}, \tag{8}$$

where $\mathbf{M}_{\mathbf{q}} = \mathbf{I}$, $\mathbf{C}_{\mathbf{q}} = (\xi\mathbf{I} + \zeta\Lambda)$, and $\mathbf{K}_{\mathbf{q}} = \Lambda$ are now all diagonal[†], and $\Phi^{\mathrm{T}}\mathbf{F}$ is the modal force. The decoupling shown in Equation (8) brings a tremendous speed-up in the numerical computation. The essence of modal warping technique is to decompose a large deformation into a series of small deformations for which the above decoupling procedure can apply to; The details are presented in Section 3.3.

### 3.2. Modal Rotations

To develop a modal warping technique for thin shells, we must develop a procedure to represent the rotational component of deformation in terms of $\mathbf{q}(t)$. More specifically, we need an equation of a similar form as Equation (7), but this time for $\mathbf{w}(t)$, the $3n$-dimensional vector formed by concatenating all the 3D rotation vectors of the mesh nodes. In the case of 3D solids, the curl $\frac{1}{2}\nabla \times \mathbf{u}$ gives the rotational component of the deformation. However, this curl-based rotation capturing is not applicable to a shell because the differentiation involved in the curl operation should not be done over free 3D space but be restricted to the 2D domain occupied by the shell.

In this section, we develop a novel procedure to calculate the rotational component of deformation. It is based on the Jacobian of triangle orientation. Imagine a triangle labelled $A$ undergoes the deformation shown in Figure 2, in which $\mathbf{x}_i$ represent the vertices in the undeformed state, $\mathbf{u}_i$ represent the displacements occurred, and $\mathbf{a}_i$ represent the position of the vertices after the deformation. Let $\omega_A$ be the 3D rotation vector that represents the orientational change occurred to $A$ by this deformation; $\omega_A$ encodes the rotation that is made around the unit axis $\omega_A/\|\omega_A\|$ by the angle

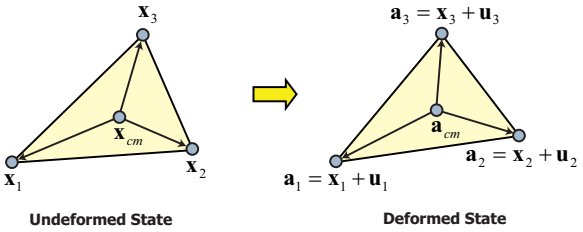


**Figure 2:** *Finding the rotational component $\omega_A(\mathbf{u}_A)$ of a triangle, the deformation of which is given with $\mathbf{u}_A = [\mathbf{u}_1^{\mathrm{T}}|\mathbf{u}_2^{\mathrm{T}}|\mathbf{u}_3^{\mathrm{T}}]^{\mathrm{T}}$.*

$\|\omega_A\|$. This rotation vector must be a purely geometric function $\omega_A(\mathbf{u}_A)$ of the displacements of the three triangle nodes, $\mathbf{u}_A = [\mathbf{u}_1^{\mathrm{T}}|\mathbf{u}_2^{\mathrm{T}}|\mathbf{u}_3^{\mathrm{T}}]$.

The problem of finding the rotation occurred to the triangle shown in Figure 2 can be formulated as:

$$\arg\min_{\mathbf{R}_A} \sum_{i=1}^{3} \left\|\mathbf{R}_A(\mathbf{x}_i - \mathbf{x}_{cm}) - (\mathbf{a}_i - \mathbf{a}_{cm})\right\|^2, \tag{9}$$

where $\mathbf{R}_A$ is the $3 \times 3$ rotation matrix, $\mathbf{x}_{cm} = \frac{1}{3}\sum_i \mathbf{x}_i$, and $\mathbf{a}_{cm} = \frac{1}{3}\sum_i \mathbf{a}_i$. Unfortunately, differentiating this equation is difficult because it contains the rotation matrix. When the rotation is infinitesimal, the rotation matrix $\mathbf{R}_A$ can be approximated by $(\mathbf{I} + \omega_A\times)$ [Gol83], where $\mathbf{z}\times$ denotes the standard skew symmetric matrix of vector $\mathbf{z}$. Then Equation (9) can be written as

$$\arg\min_{\omega_A} \sum_{i=1}^{3} \left\|(\mathbf{I} + \omega_A\times)(\mathbf{x}_i - \mathbf{x}_{cm}) - (\mathbf{a}_i - \mathbf{a}_{cm})\right\|^2. \tag{10}$$

If we use the notations $\mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{cm}$ and $\mathbf{q}_i = \mathbf{a}_i - \mathbf{a}_{cm}$, the above equation can be simplified into

$$\arg\min_{\omega_A} \sum_{i=1}^{3} \|\mathbf{p}_i + \omega_A \times \mathbf{p}_i - \mathbf{q}_i\|^2. \tag{11}$$

Equating the derivative of the objective function of Equation (11) with respect to $\omega_A$ to zero, we obtain

$$\Big(\sum_i \mathbf{p}_i \times \mathbf{p}_i \times\Big)\omega_A = -\sum_i \mathbf{p}_i \times \mathbf{q}_i. \tag{12}$$

Here, $\mathbf{q}_i$ and thus $\omega_A$ are functions of the displacement $\mathbf{u}_A$. Differentiating both sides of Equation (12) with respect to $\mathbf{u}_A$ and evaluating the derivative for the undeformed state, we obtain the $3 \times 9$ Jacobian matrix

$$\frac{\partial\omega_A}{\partial\mathbf{u}_A}\Big|_{\mathbf{0}} = -\Big(\sum_i \mathbf{p}_i \times \mathbf{p}_i \times\Big)^{-1}\big[\mathbf{p}_1 \times \big|\, \mathbf{p}_2 \times \big|\, \mathbf{p}_3 \times\big]. \tag{13}$$

Equation (13) is derived under the assumption that the deformation contains a small rotational component. The modal warping procedure introduced in Section 3.3 takes the approach of decomposing a large rotational deformation into a number of small rotational deformations so that the above result can be applied.

[†] We take the commonly adopted assumption (*Rayleigh damping*) that $\mathbf{C} = \xi\mathbf{M} + \zeta\mathbf{K}$, where $\xi$ and $\zeta$ are scalar weighting factors.

With the Jacobian given in Equation (13), we can approximate $\omega_A(\mathbf{u}_A)$ with first-order Taylor expansion

$$\omega_A(\mathbf{u}_A) = \omega_A(\mathbf{0}) + \frac{\partial \omega_A}{\partial \mathbf{u}_A}\Big|_{\mathbf{0}} \mathbf{u}_A + O(\mathbf{u}_A^2). \tag{14}$$

Here $\omega_A(\mathbf{0})$ is zero because there is no rotation at the undeformed state. The above was the procedure to obtain the rotation vector of a triangle. We obtain the rotation vector of a mesh node by taking average of the rotation vectors of the triangles sharing the node.

Based on the above discussion, we can now assemble the Jacobians $\partial\omega_A/\partial\mathbf{u}_A$ of all the triangles to form the global matrix $\mathbf{W}$ such that $\mathbf{Wu}$ gives the $3n$ dimensional composite vector $\mathbf{w}$. Finally, expanding $\mathbf{u}(t)$ with Equation (7) gives

$$\mathbf{w}(t) \approx \mathbf{W}\Phi\mathbf{q}(t) \triangleq \Psi\mathbf{q}(t). \tag{15}$$

Both $\mathbf{W}$ and $\Phi$ are characterized by the thin shell at the undeformed state and are thus constant over time. Therefore we can precompute $\Psi$. The above equation shows that, as in the displacement given in Equation (7), we can represent local rotations of mesh nodes in terms of $\mathbf{q}(t)$. We call $\Psi$ the *modal rotation matrix*.

### 3.3. Integration of Rotational Parts

Equation (15) provides an efficient way to keep track of the rotations occurring at the shell nodes. However, such rotations are not yet reflected in the calculation of the displacement field $\mathbf{u}(t)$. Moreover, the results derived in Sections 3.1 and 3.2 hold only when rotational components of the deformation are moderately small. Both of these problems can be resolved by introducing a local coordinate frame to each mesh node.

We embed a local coordinate frame $\{i\}$ at each node $i$ such that at the initial state it is aligned with the global coordinate frame. We use the notation $\{i\}(t)$ to refer to the local coordinate frame at time $t$. Let $\mathbf{R}_i(t)$ be the rotation matrix representing the orientation of $\{i\}(t)$, and $\dot{\mathbf{u}}_i^L(t)dt$ be the differential displacement of the $i$-th node at time $t$ measured from $\{i\}(t)$. Then, the finite displacement $\mathbf{u}_i(t)$ measured from the global coordinate frame can be calculated as follows:

$$\mathbf{u}_i(t) = \int_0^t \mathbf{R}_i(\tau)\dot{\mathbf{u}}_i^L(\tau)d\tau. \tag{16}$$

The above integration must be carried out for every node. We refer the readers to [CK05] for the detailed procedure to calculate Equation (16).

For the calculation of Equation (16), we need a new governing equation rather than Equation (6) which can be solved for $\mathbf{u}^L$, the generalized displacement vector measured in the time-varying local coordinate frames. [CK05] shows that, by premultiplying $\mathbf{R}$ to both sides of Equation (6) and making assumptions on *commutativity in fine meshes* and *warped stiffness*,

$$\mathbf{M}\ddot{\mathbf{u}}^L + \mathbf{C}\dot{\mathbf{u}}^L + \mathbf{K}\mathbf{u}^L = \mathbf{R}^{\mathrm{T}}\mathbf{F} \tag{17}$$

can be obtained, where $\mathbf{K}$ is $\mathbf{K}(\mathbf{u})$ in the undeformed state and $\mathbf{R}$ is a $3\times 3$ block diagonal matrix constructed with $\mathbf{R}_i$s.

Note that (1) the rotations that occurred over time at the mesh nodes are now reflected in the result of Equation (16), and (2) since it integrates small rotations, the equations derived in Sections 3.1 and 3.2 are applicable and can simulate large rotational deformation of shells.

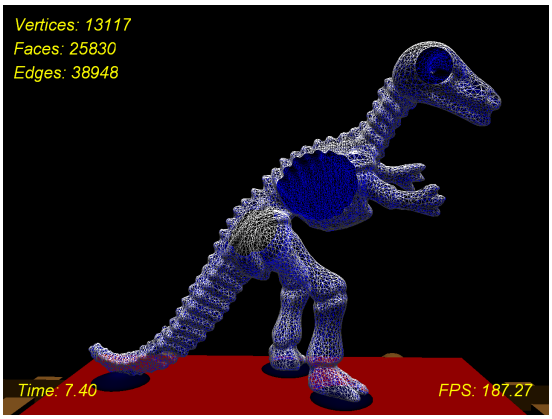


**Figure 3:** *Real-time deformation of a large mesh.*

## 4. Experiments

The proposed technique was implemented into an Autodesk MAYA plugin that runs in Microsoft Windows$^{XP}$ environment. To obtain the $m$ dominant eigenvalues of large sparse square matrices and corresponding eigenvectors, we used the built-in C++ math function `eigs` in MATLAB. The Jacobians of the energy functions were calculated symbolically with Maple. All experiments described in this section were performed on a PC with an Intel Pentium D 3.46GHz processor, 2GB memory, and an *n*VIDIA GeForce FX 7900 GTX graphics card. In all experiments, we fixed the time step size to $h = 1/30$ second but we did not encounter any instability problem.

### 4.1. Large Mesh Test

We simulated deformation of a 3D dinosaur shape shell which consists of 25,830 triangles (13,117 vertices and 38,948 edges) with the proposed technique. The simulation was done with four modes, which needed about 18 seconds of pre-computation. The model was excited by moving the basis with non-uniform velocities. To achieve real-time performance, we employed a vertex program on a programmable graphics hardware as in [CK05]. Fig. 3 shows a snapshot taken during this experiment. The simulation ran at about 200 fps.

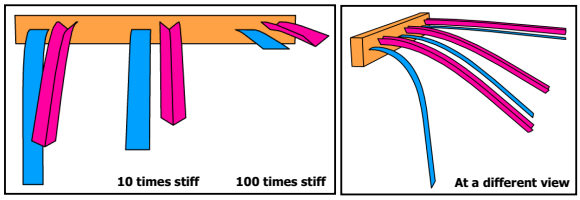


**Figure 4:** *Simulation of flat and V-beams deforming in the gravity field.*

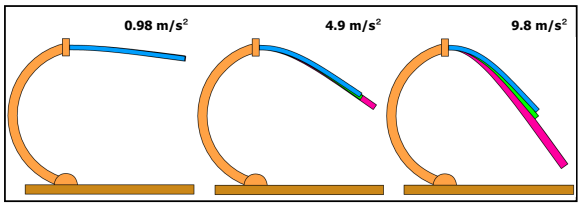


**Figure 5:** *A V-beam deformed by linear modal analysis (red), by modal warping for thin shells (blue), and by discrete shells (green) under gravities of different magnitudes.*

### 4.2. Flat Beam and V-Beam Test

In this experiment, we tested the bending of flat beams and V-beams in the gravity field. Fig. 4 shows three pairs of results generated by the proposed technique using eight modes. Each pair consists of a flat beam and a V-beam of the same flexural stiffness for side-by-side comparison. From left to right, the pairs have increasing flexural stiffness; The middle pair is ten times as stiff as the leftmost one, and the rightmost pair is hundred times as stiff as the leftmost one. The figure shows that the V-beams have different bending behaviors from the flat beams, and the difference is more dramatic at smaller flexural stiffness.

### 4.3. Comparison to Other Methods

This experiment is to compare the results generated by linear modal analysis (LMA), modal warping for thin shells (MWTS), and nonlinear discrete shells (DS) of Grinspun et al. [GHDS03]. We ran the three methods to deform a V-beam under different gravities. The simulations with LMA and MWTS were both performed using eight modes. For running DS, we employed explicit integration and used the time step size $h = 1/30,000$ second for numerical stability. Fig. 5 shows the snapshots taken at the equilibrium states.

Fig. 6 (a) shows the plot of the relative $L_2$ displacement field error versus gravitational magnitude. We took the result produced by DS as the ground truth.[‡] The error of MWTS is smaller than that of LMA although both of them increase as the gravitational magnitude increases. Fig. 6 (b) is the plot of the relative area change with respect to the initial area. It shows that the area change in MWTS is almost identical to that in DS. Even though Fig. 6 (a) shows MWTS produces non-negligible $L_2$ displacement field errors, it was not easy to visually distinguish between the results produced with MWTS and DS, unless the results were seen overlayed.

[‡] MWTS and DS use the same energy function, but MWTS makes some simplifications/assumptions that DS does not make. Therefore MWTS cannot be more accurate than DS. It is the *price* MWTS should pay for being a real-time algorithm. In principle, the result of nonlinear FEM simulation based on shell theory should be taken as the ground truth. But we did not take this approach because (1) implementation of nonlinear FEM simulation based on shell theory was too difficult, and (2) the emphasis of the present work is not on the accuracy of dynamic formulation but on the development of real-time artifact-free simulation technique for a *given* dynamic formulation. For this purpose, comparing the result of MWTS with that of DS was considered good enough.

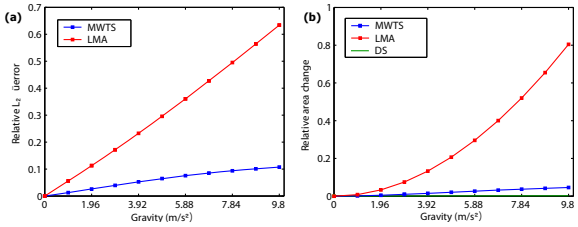


**Figure 6:** *Error analysis of the V-beam shown in Fig. 5. (a) The relative $L_2$ displacement field error. (b) The area change with respect to the initial area; The area change of DS is almost zero so the curve is almost undistinguishable from the horizontal axis.*

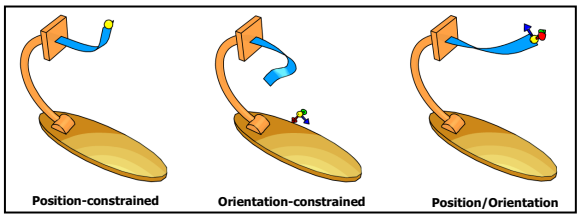


**Figure 7:** *A flat beam manipulated with a position constraint (left), with an orientation constraint (middle), and with both position and orientation constraints (right). The position constraints are represented by yellow spheres and the orientation constraints are represented by RGB axes.*

### 4.4. Manipulation Constraints

The manipulation constraints introduced in modal warping for 3D elastic solids [CK05] can be applied to modal warping for thin shells. Fig. 7 shows the snapshots taken while

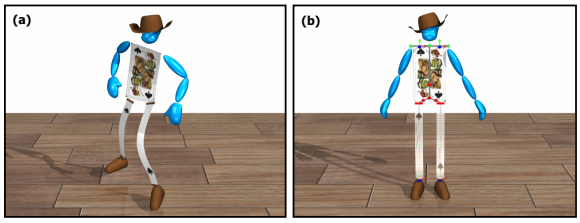


**Figure 8:** *Constraint-driven animation of a character consisting of four thin shells (the hat, body, and two legs).*

a flat beam is manipulated with position/orientation constraints. The manipulation constraints can be used to animate a deformable character composed of thin shells. We simulated a character whose upper body and legs are made of thin shells (Fig. 8 (a)). As the character makes a dance motion, the shells were made to make passive dynamic deformations, excited by the gross body motion of the character. As shown in Fig. 8 (b), the shells were attached to the skeleton by position/orientation constraints (the RGB axes).

## 5. Conclusion

In this paper, we proposed a real-time simulation technique for thin shells. We formulated dynamics of thin shells using the energy functions proposed in [GHDS03]. Then, we made modifications to the modal warping technique, which was originally proposed for 3D elastic solids [CK05], so as to be used for simulating thin shells. The task involved development of a novel procedure to find the rotational components of deformation in terms of the modal amplitudes. Also, we showed that the manipulation constraints introduced in [CK05] can be extended to thin shells. The proposed technique ran stably even when the time step size was fixed to $h = 1/30$ second, and produced visually convincing results.

Although the current implementation is done with the energy functions introduced in Section 2, it can work for any given dynamic formulation. We expect the proposed technique will prove useful in broad application areas, including computer games and character animation.

## Acknowledgments

This work was supported in part by Seoul R&BD Program, Ministry of Information and Communication under the Information Technology Research Center (ITRC) support program, the Research Grant of Kwangwoon University in 2007, Ministry of Science and Technology under National Research Laboratory (NRL) grant M10600000232-06J0000-23210, the Brain Korea 21 Project, and Automation and Systems Research Institute at Seoul National University.

## References

[BJ05] BARBIĆ J., JAMES D. L.: Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2005) 24*, 3 (2005), 982–990.

[CK05] CHOI M. G., KO H.-S.: Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Transactions on Visualization and Computer Graphics 11*, 1 (2005), 91–101.

[GHDS03] GRINSPUN E., HIRANI A. N., DESBRUN M., SCHRÖDER P.: Discrete shells. In *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation* (2003), pp. 49–54.

[GM97] GIBSON S., MIRTICH B.: *A Survey of Deformable Modeling in Computer Graphics*. Tech. Rep. TR-97-19, Mitsubishi Electric Research Lab., Cambridge, MA, 1997.

[Gol83] GOLDSTEIN H.: *Classical Mechanics*. Addison-Wesley, 1983.

[GZ68] GREEN A. E., ZERNA W.: *Theoretical Elasticity*. Oxford University, 1968.

[HSO03] HAUSER K. K., SHEN C., O'BRIEN J. F.: Interactive deformation using modal analysis with constraints. In *Proc. Graphics Interface* (2003), pp. 247–255.

[JF03] JAMES D. L., FATAHALIAN K.: Precomputing interactive dynamic deformable scenes. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2003) 22*, 3 (2003), 879–887.

[MDM*02] MÜLLER M., DORSEY J., MCMILLAN L., JAGNOW R., CUTLER B.: Stable real-time deformations. In *Proc. ACM SIGGRAPH Symp. Computer Animation 2002* (2002), pp. 49–54.

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *Computer Graphics (Proc. ACM SIGGRAPH '87) 21*, 4 (1987), 205–214.