ICAT 2003
December 3-5, Tokyo, JAPAN

Real-time Method for Animating Elastic Objects' Behaviors Including Collisions

Takafumi WATANABE*1 Jun OHYA*1
Jun KURUMISAWA*2,*3 Yukio TOKUNAGA*4

*1 Graduate School of Global Information and Telecommunication Studies, Waseda University
1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo 169-0051, Japan
*2 Faculty of Policy Informatics, Chiba University of Commerce
1-3-1 Konodai, Ichikawa-shi Chiba 272-8512 Japan
*3 Global Information and Telecommunication Institute, Waseda University
1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo 169-0051, Japan
*4 Faculty of Engineering, Shibaura Institute of Technology
307 Oazafukasaku, Minuma-ku, Saitama 337-8570, Japan
t_watanabe@ruri.waseda.jp

Abstract

We present a new real-time method that can animate a moving elastic object that could collide with other objects in a virtual environment. For simulating physical motions of elastic objects, we exploit a Boundary Element Method (BEM), which can achieve efficient computations and desired deformation as opposed to spring models and a Finite Element Method. In case that real-time processes are required for the BEM, only the physical simulation cannot achieve animations; therefore, this paper proposes a 2D model that is combined with the BEM. Since it is difficult to animate 3D elastic objects in real-time, we propose a method that approximates the 3D motions based on the 2D model. The effectiveness of the proposed method is demonstrated by the experiments in which the dynamical behaviors of a jelly are reproduced in a virtual environment in real-time.

Key words: elastic object, boundary element method, real-time animation, physical simulation

1. Introduction

We study a reproduction method of behaviors of non-rigid objects in virtual environment. To conduct such a research project, it is essential to achieve a real-time method for animating non-rigid objects' dynamical behaviors. The physical simulation is one of the most important technologies needed for generating such animations automatically. As detailed in the following, physical simulations are classified into off-line type (non-real-time) physical simulations and real-time physical simulations.

Off-line type physical simulations are used for applications such as creating digital cinemas. The off-line type physical simulations often use super computers for processing huge amount of computation so that the accuracy and visual sensation of the animation results can be pursued. Sometimes, each frame of a digital cinema is manually created by utilizing the animation result computed.

The real-time physical simulations are frequently used for interactive applications such as video games and training simulators. In these areas, the non-rigid object is a challenging target for physical simulations. In general, physical simulations for non-rigid objects tend to be inefficient for computations if the accuracy and visual sensation of the animation result are pursued. Video games and other interactive systems, however, are required to respond to the user's input in real-time; thereby, the physical simulation is also required to be performed in real-time For this reason, it is very difficult to solve the above-mentioned dilemma of the physical simulation.

The non-rigid object that is being researched includes elastic objects, clothes [3] and human faces [4]. This paper studies elastic objects. Elastic objects are studied in surgery simulations, in which human internal organs are modeled as elastic objects. One of the main research topics in surgery simulations is to simulate the dynamical deformations of internal organs using elastic object models. Thus, technologies that animate elastic objects that do not change their positions can be seen, but the technology that can animate elastic objects that change their positions and are deformed has not yet been developed. This paper studies a real-time method that can animate the elastic objects that move and are deformed due to kinematic effects such as collisions. More specifically, the model that can be applied to the situation in which an elastic object falls down from some height to a flat plane and bounces is addressed.

As another issue for animating elastic objects, how to give physical parameters to elastic objects is a problem to be solved. In case of rigid bodies, only a few parameters such as mass and moment of inertia need to be controlled. On the other hand, elastic objects in general need many physical parameters. Many physical parameters make it difficult to simulate desired deformations of elastic objects. Ideally, the number of physical parameters should be small so as to facilitate analyzing the relationship between each physical parameter value and deformation results. As detailed in Section 2, a boundary element method (BEM) is used as the simulation method that satisfy the above-mentioned conditions.

This paper is organized as follows. Section 2 compares possible methods that can be applied to animating elastic objects and claims that the BEM is the best choice for the purpose of this paper. In section 3, the fast computation method for a BEM is described. In section 4 and section 5, based on the above methods, we propose a 2D animation model and a 3D animation model for real-time computation, respectively. In section 6, the effectiveness of the proposed method is specified by the experiments in which the dynamical behaviors of a jelly are reproduced in real-time. Section 7 concludes this paper.

2. Technologies Relevant to Elastic Simulations

Methods for simulating the deformations of elastic objects include spring models, a finite element method (FEM), and a boundary element method (BEM).

Spring models approximate the shape of the object by a wire frame model whose vertices are connected by springs. Spring models are featured by their efficient computations and adaptability to different applications. On the other hand, since the spring model is a bottom-up approach in which the entire deformations of an elastic object is obtained from computing the kinematic relationship between each spring and the connected vertices, different values for physical parameters such as the mass of each vertex and the stiffness of each spring cause different deformations of the elastic object; in other words, generally speaking, it is very difficult to predict the resultant deformations. Furthermore, the spring model tends to lead to undesirable vibrations in animating elastic objects' behaviors. One of the application areas of the spring model is surgery simulations, because surgery simulations require the function that cuts the elastic objects that model human internal organs. There is a spring model based approach that can deal with elastic objects' positional changes [2][7]. This approach aims at achieving the real-time computation and desired deformations by studying how to position the vertices in the elastic object.

The finite element method (FEM) is one of numerical analysis methods and is widely used in different engineering areas such as the fracture mechanics [5], because the FEM can basically achieve desired deformations. The FEM discretizes the object by partitioning not only its surface but also the inside of the object into polygons. Although only the surface of an elastic object should be displayed, the FEM computes the vertices inside the object also. This results in computational inefficiencies; that is, it can be said that the FEM is not appropriate for real-time applications.

The boundary element method (BEM) is a numerical analysis method used in engineering areas. The BEM is similar to the FEM, but the BEM can overcome the computational inefficiency problem of the FEM as described in the following. Since the BEM discretizes only the surface of an elastic object (does not discretize inside the object), the computation efficiency is much higher than the FEM. In fact, a fast algorithm that can be applied to linear elastic objects was developed [1]. Using this algorithm, it is possible to achieve real-time simulations for linear elastic objects. Another advantage of the BEM is that the BEM is a top-down approach in which the deformation of an elastic object is simulated by solving a boundary integral equation. That's why the BEM can achieve predicted deformations simply by giving the physical parameter values to the elastic object model based on the principle of physics.

As described in the subsequent sections, this paper utilizes the BEM for animating the dynamical behaviors of elastic objects.

3. Boundary Element Method

This section outlines the BEM and D. L. James et al.,'s fast BEM algorithm [1]. Then, the BEM based method proposed in this paper is explained.

3.1 Boundary Integral Equation

The BEM solves the following boundary integral equation, where i and j deno independent coordinate axes respectively.

$$C_{ij}(P)u_j(P)+\int_S T_{ij}(P,Q)u_j(Q)\,dS(Q) = \int_S U_{ij}(P,Q)t_j(Q)\,dS(Q)+\int_V U_{ij}(P,q)b_j(q)dV(q) \tag{1}$$

Equation (1) is used for obtaining the displacement at the pint P on the boundary (surface) S of the elastic object (Fig. 1), where Q is another point on the boundary, C_{ij} is a constant that is determined by the shape and material of the boundary, and T_{ij} and U_{ij} are kernel functions based on Navier equation. The second term of the right side of Eq. (1) represents the internal body force, where $b_{ij}(p)$ is a body force function, and q is a point on the surface or inside of the object.

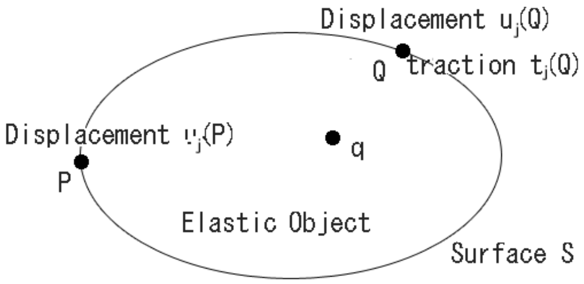

Figure 1. Boundary integral equation

Kernel functions T_{ij} and U_{ij} are represented by the following equations if the elastic object is 2-dimensional linear elastostatic.

$$U_{ij}(P,Q)=\frac{1}{8\pi(1-\nu)\mu}\left\{(3-4\nu)\ln\left(\frac{1}{r}\right)\delta_{ij}+r_{,i}r_{,j}\right\}$$
$$T_{ij}(P,Q)=-\frac{1}{4\pi(1-\nu)r}\left[\{(1-2\nu)\delta_{ij}+2r_{,i}r_{,j}\}\frac{\partial r}{\partial n}-\left(1-2\nu\left(r_{,i}n_j-r_{,j}n_i\right)\right)\right] \tag{2}$$

where U_{ij} and T_{ij} are the displacement and traction for the j direction at the point Q when a unit force for the i direction is applied to the point P in the infinite plate, respectively. In Eq. (2), ν is Poisson's ratio, distance between the points P and Q, and r_i and r_j are the partial derivatives of r for the i and j directions, respectively. By discretizing Eq. (1), the following equation is obtained.

$$\mathbf{HU}=\mathbf{GT}+\mathbf{B} \tag{3}$$

where $\mathbf{U}$ and $\mathbf{T}$ are the matrices that contain the displacements and tractions, respectively, $\mathbf{H}$ and $\mathbf{G}$ are coefficient matrices, and $\mathbf{B}$ is the internal body force matrix. By shifting unknowns for the displacements and tractions to the left side of the equation, and shifting knowns to the right side, Eq.(3) is converted to Eq. (4). Note that our problem for obtaining the deformation of the elastic object is to solve the simultaneous equations indicated in Eq. (4).

$$\mathbf{AX}=\mathbf{Z}+\mathbf{B} \tag{4}$$

where all the components of $\mathbf{X}$ are unknowns, all the components of $\mathbf{Z}$ are knowns, and $\mathbf{A}$ is the coefficient matrix obtained from the above-mentioned shift operations.

3.2 Fast Algorism

The fast BEM algorithm proposed by D. L. James et al. [1] can be used in case that no internal body force is applied to the elastic object. Since this paper deals with the case in which elastic objects change their positions, the values of the unknowns (displacements and tractions) could be changed at each time instant. In this case, Eqs (3) and (4) need to be reconstructed, and the simultaneous equations in Eq. (4) needs to be solved again. This yields much computation. D. L. James et al,'s method can avoid this problem.

In Eq. (4), suppose that $\mathbf{A}$ is changed to $\mathbf{A}'$ according to the changes in the boundary conditions. $\boldsymbol{\delta}\mathbf{A}$ be the difference between A and A'. Then, we have the following equation.

$$\mathbf{A}'=\mathbf{A}+\boldsymbol{\delta}\mathbf{A} \tag{5}$$

where there are $\boldsymbol{\delta}\mathbf{A}$'s columns in which all the components are zero. Then, let $\boldsymbol{\delta}\mathbf{A_C}$ be the matrix that is obtained from removing the all zero columns from $\boldsymbol{\delta}\mathbf{A}$. By using the n by s matrix $\mathbf{E_C}$ whose components are either 1 or 0, Eq. (5) is rewritten as follows.

$$\mathbf{A}'=\mathbf{A}+\boldsymbol{\delta}\mathbf{A_C}\mathbf{E_C}^{\mathbf{T}} \tag{6}$$

Next, the following equation is considered, where $\mathbf{A_S}$ is an n by n square matrix.

$$\mathbf{A_S}=\mathbf{A_0}+\mathbf{RS}^{\mathbf{T}} \tag{7}$$

where $\mathbf{R}$ and $\mathbf{S}$ are n by s matrices. The inverse matrix of $\mathbf{A_S}$ is represented by the following equation according to the Sherman-Morrison-Woodbury (SMW) formula.

$$\mathbf{A_S}^{-1}=\mathbf{A_0}^{-1}-\mathbf{A_0}^{-1}\mathbf{R}\left[\mathbf{I}+\mathbf{S}^{\mathbf{T}}\mathbf{A_0}^{-1}\mathbf{R}\right]^{-1}\mathbf{S}^{\mathbf{T}}\mathbf{A_0}^{-1} \tag{8}$$

where $\mathbf{I}$ is the $s\times s$ identity matrix. By substituting Eq. (6) into Eqs. (4) and (8), the following equation is obtained.

$$\mathbf{A}'^{-1}\mathbf{Z}=\left(\mathbf{A}^{-1}\mathbf{Z}\right)-\left(\mathbf{A}^{-1}\boldsymbol{\delta}\mathbf{A_C}\right)\left[\mathbf{I}+\mathbf{E_C}^{\mathbf{T}}\left(\mathbf{A}^{-1}\boldsymbol{\delta}\mathbf{A_C}\right)\right]^{-1}\mathbf{E_C}^{\mathbf{T}}\left(\mathbf{A}^{-1}\mathbf{Z}\right) \tag{9}$$

By solving Eq. (9), the displacements and tractions of the elastic object can be obtained. Note that this computation is very efficient, because Eq.(9) requires only the computation of the inverse of the s by s matrices and some matrix multiplications.

3.3 Inertia force

The method described in Section 3.2 assumes that no internal body force is applied to the elastic object. Simulating the inertial force as the internal body force is essential to our method proposed in this paper. The inertia force varies according to time; therefore, the matrix $\mathbf{B}$ in Eq. (3) needs to be reconstructed at each time instant. This reconstruction results in inefficient computations. A real-time method that can avoid this problem needs to be established.

When the inertial force is applied to the elastic object, the second term of the right side of Eq. (1) is written as

$$B_i^P \equiv \frac{1+\nu}{4\pi E}\int_S r\left(2\ln\frac{1}{r}-1\right)\left[b_i n_m r_{,m} - \frac{b_m r_{,m} n_i}{2(1-\nu)}\right] dS(Q) \quad (10)$$

If the inertia force varies, the values of b_i change. Thus, Eq. (10) can be rewritten as follows.

$$B_i^P = b_i \beta_1^P + b_1 \beta_{2\,i}^P + b_2 \beta_{3\,i}^P \quad (11)$$

where

$$\begin{aligned}
\beta_1^P &= \frac{1+\nu}{4\pi E}\int_S r\left(2\ln\frac{1}{r}-1\right) n_m r_{,m} dS(Q) \\
\beta_{2\,i}^P &= -\frac{1+\nu}{4\pi E}\int_S r\left(2\ln\frac{1}{r}-1\right)\frac{r_{,1} n_i}{2(1-\nu)} dS(Q) \\
\beta_{3\,i}^P &= -\frac{1+\nu}{4\pi E}\int_S r\left(2\ln\frac{1}{r}-1\right)\frac{r_{,2} n_i}{2(1-\nu)} dS(Q)
\end{aligned} \quad (12)$$

From these equations, Eq. (4) is rewritten as

$$\mathbf{AX} = \mathbf{Z} + \mathbf{B} = \mathbf{Z} + b_1 \mathbf{B_1} + b_2 \mathbf{B_2} \quad (13)$$

Note that Eq. (13) can be applied to Eq. (9) so that the BEM with the inertia force can be computed efficiently.

4. 2D Animation

As can be seen in Eqs. (1) and (2), the BEM based elastic simulation does not include a temporal term. This fact causes the following two problems. (i) Temporal simulations of elastic objects cannot be performed. In other words, animating elastic objects cannot be achieved only by the BEM based elastic simulation. (ii) The BEM based elastic simulation can deal with only static objects. This means that the simulation can be performed only when the internal and external forces are balanced.

4.1 2D Animation Model

In order to solve the first problem (i), a rigid body simulation is combined with the elastic simulation; that is, the rigid body simulation computes the change in the object's position so as to make it possible to animate the object's behaviors. For example, in the case that an elastic object is thrown at a wall, only the rigid body simulation is performed till the elastic object collides with the wall. At the time that the elastic object collides with the wall, the deformation is simulated by the elasticity simulation, and the position is changed by a rigid body simulation simultaneously. If the elastic object bounces off the wall, the elastic simulation is stopped. By repeating the above-mentioned processes, animating the elastic object's behaviors is completed. In order to solve the second problem (ii), the resultant of the forces applied to the elastic object must be set to zero. In the above-mentioned example, in which the object collides with the wall, this problem occurs on the collision. The only external force in this example is the reaction caused by the wall. Thus, if F is defined as the reaction force, the equation of motion can be represented by

$$m\frac{d^2x}{dt^2} = F \quad (14)$$

where m and x are the mass and the barycentric coordinate of the elastic object, respectively. Since the resultant is not zero in this condition, the elasticity simulation cannot perform the deformation. In order to perform the elasticity simulation, the force F' that satisfies the following equation is needed.

$$0 = F' + F \quad (15)$$

Equation (15) is converted to the following representation.

$$0 = -m\frac{d^2x}{dt^2} + F \quad (16)$$

Equation (16) corresponds to the situation in which the resultant is equal to zero (Fig. 2). By comparing Eq. (15) with Eq. (16), the inertia force is appropriate for F'.

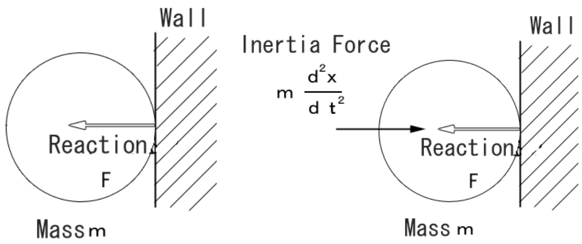

Figure 2: forces at the time that an elastic object collided with a wall.

Here the strength of the inertia force must be discussed. The inertia force is the acceleration generated by F.

F is the force that is generated by the wall that pushes back the elastic object. The force that pushes back depends on the volume V, which corresponds to the elastic object's volume that would pierce through the wall if the object were not deformed due to the collision (Fig. 3). We assume that the force is proportional to V.

$$F \propto V \tag{17}$$

where the constant of proportion can change the stiffness of the elastic object.

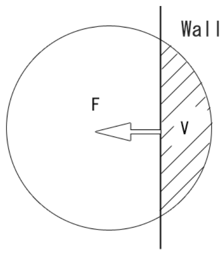

Figure 3: Volume V. This elastic object is not deformed.

4.2 Physical Parameters

The elasticity simulation that uses our 2D animation model has two physical parameters: Poisson's ratio and Young's modulus.

Young's modulus of an elastic object is the constant of proportion that expresses the ratio between a displacement and the traction generated by the displacement. A large value of the Young's modulus gives a solid object that bounces much, while a small value gives a soft object that bounces little.

Poisson's ratio is the ratio of the transversal contraction to the longitudinal extension. A large value of Poisson's ratio (at maximum, 0.5) gives a rubber-like incompressible object, while a small value gives a sponge-like compressible object. As shown in Fig. 4, different values for the physical parameters give different deformations of elastic objects.

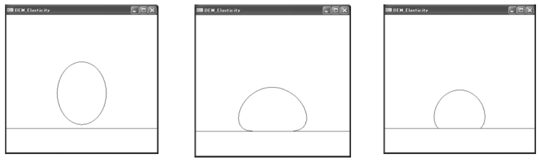

Figure 4: Left figure is before an elastic object collides with a floor, and Poisson' ratio 0.5 (middle) and 0.01 (right) after collision with a floor.

4.3 Simulation Result

Figure 5 shows an animation created by our 2D animation model, where an elastic circle falls from some height to a flat floor. Using the Celeron 2.0GHz + Open GL, the computation speed is faster than 60 frames per second.

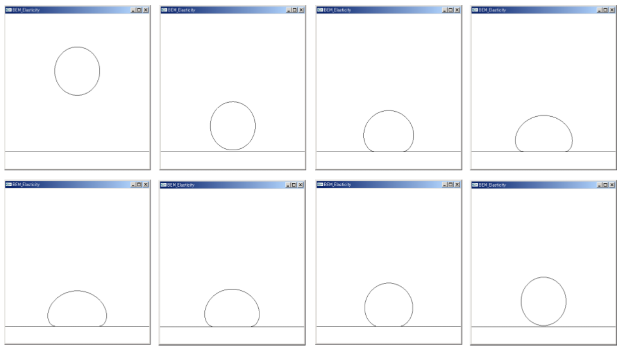

Figure 5: Animation of a 2D elastic object.

5. 3D Animation Model

One of the most serious problems for animating elastic objects in 3D is computation inefficiency. The amount of computation is proportional to the number of vertices composing the elastic object. In general, the number of vertices in 3D objects is larger than that for 2D objects; thereby, the computation cost for 3D objects is much larger than 2D objects. In this paper, we propose a method that approximates the 3D deformation by the 2D model.

5.1 Approximate Model

In our approximation model, a 3D elastic object is represented by a set of 2D planes. More specifically, by applying our 2D animation model to each of cross sections of a 3D object, the animation result can be as good as the result obtained from the 3D animation model. However, if the number of the cross sections is large, the computation gets inefficient. Therefore, this paper assumes that the number of the cross sections is two. In this case, the shape of 3D objects is limited to simple shapes such as spheres and cylinders, but the computation cost is very small.

Figure 6 shows the two cross sections of the cylindrical object. The procedure consists of the following three steps.

① Compute the mean coordinates of the contact (collision) area at the time the elastic object collides.

② Determine the first cross section that passes the mean contact (collision) point and the centroid of the elastic object.

③ Determine the second cross section that is perpendicular to the first cross section.

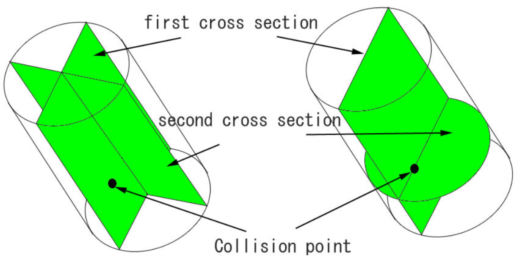

Figure 6: The 2D planes can be selected two types. We select the plane that has more collision points.

5.2 Approximation Method

After the displacements in the cross sections are computed, the 3D deformation of the elastic object is approximated based on the computed displacements (Fig. 7). Each vertex of the 3D elastic object is projected to the x' - y' coordinate system in the 2D cross section as the point A in Fig. 7. Then, from the displacements obtained in the cross section, the displacements for the x' and y'' directions in the 3D object are computed.

To reduce the computation cost, only two vertices for one direction are used for approximating the displacement. As shown in Fig. 7, the two vertices are the point B and C, which are the points at which the boundary of the cross section and the line that passes the point A and is parallel to the x' axis. By giving weights to the displacements at B and C, the displacements of the 3D object is approximated. Suppose that r_1 and r_2 are the distance between A and B and the distance between A and C, respectively. If the displacements for the x' direction at B and C are u_1 and u_2, respectively, the 3D displacement $u_{x'}$ for the x' direction is given by

$$u_{x'} = \frac{r_2{}^{\alpha}}{r_1{}^{\alpha} + r_2{}^{\alpha}} u_1 + \frac{r_1{}^{\alpha}}{r_1{}^{\alpha} + r_2{}^{\alpha}} u_2 \qquad (18)$$

Note that in Eq. (18), if α is smaller than 1, the displacement is small, while if α is large, the displacement is large. It can be said that α should be between 1 and 2 (Fig. 8).

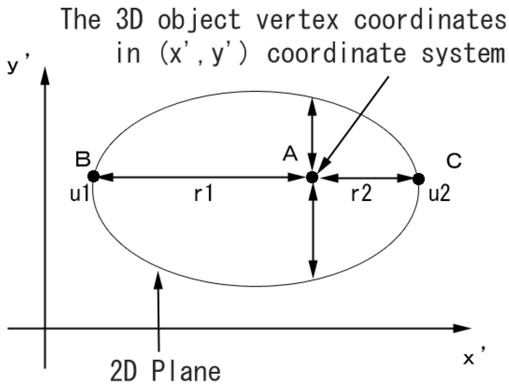

Figure 7: the displacements on the 2D planes approximate the 3D deformation.

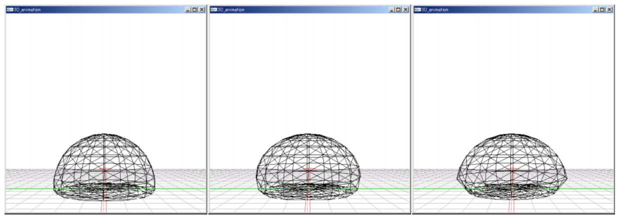

Figure 8: 3D deformations are changed depending on a value of α (α = 0.1, 1.5, 10).

5.3 Simulation results

Figure 9 shows an animation that an elastic ellipsoid falls some height to a flat floor. The elasticity, which are specified by only three physical parameters: Young's modulus, Poisson's ratio, and internal friction, can reproduce desired deformation of the elastic object.

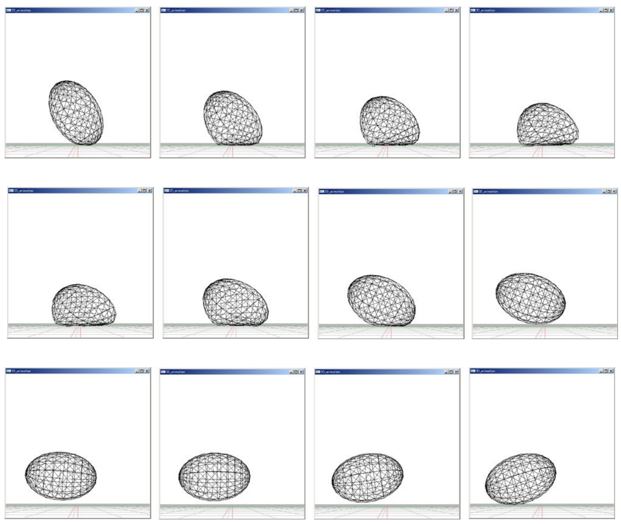

Figure 9: This animation is that an elastic ellipsoid falls to a flat floor.

6. Reproduction of Actual Behaviors of a Jelly

Dynamical behaviors of a jelly is video-taped and reproduced using the 3D animation model. The simulation in section 5 animates simple behaviors of 3D elastic objects such as a soft ball. On the other hand, it is very difficult to reproduce behaviors of the jelly because of complicated motions such as vibrations.

At the time that a jelly falls to a flat floor and touches the floor, the external and internal forces to a jelly are only two forces from gravity and the floor. Normally, the only force from the floor is the reaction to the object. However, only the reaction is not enough to reproduce jelly's behaviors. Therefore, in our model, by assuming that the forces are the reaction and a force sticking to the floor, behaviors of a jelly can be reproduced.

Soft objects such as a jelly may be deformed with a large amount; thereby, the soft objects could pierce through the floor (Fig.10). To deal with this issue, the object's volume that pierces through the floor is assumed to stay within the collision plane (floor); the simulation is performed once more for the object that is forced to stay within the floor. This method correctly simulates a jelly's deformation.

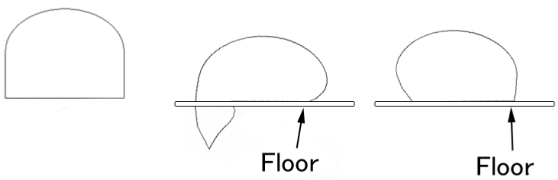

Figure 10: Left figure is an elastic object is not deformed, and the first elasticity simulation (middle) and once more (right).

The upper row of Fig.11 and Fig.12 show a scene in which a jelly object falls to a flat floor. The lower row of Fig.11 and Fig.12 show the animation created by our method. By using the Celeron 2.0 GHz (OpenGL), the computation speed is over 60 frames per second.

7. Conclusion

This paper proposes a real-time animation model for elastic objects. The proposed model is based on the boundary element method (BEM) so that elastic objects' dynamical behaviors including collisions can be animated. The proposed 2D model combines the BEM based elastic simulation with a rigid body simulation so that real-time animations can be achieved. In the proposed 3D approximation model, our 2D model is applied to two cross sections of a 3D elastic objects so that real-time 3D animations are realized. By giving different values for the physical parameters used in the BEM, different dynamical behaviors can be animated.

Remaining issues include a 3D model that does not utilize approximations. More complicated shaped elastic objects should be studied.

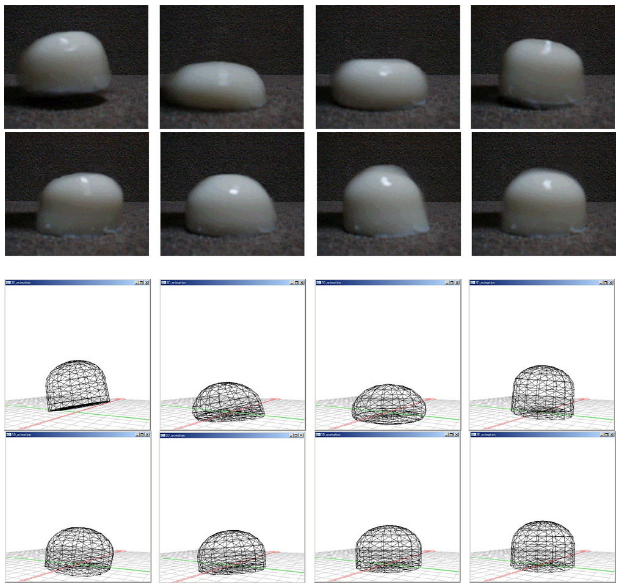

Figure 11: The upper row shows behaviors of jelly, and the lower row shows the animation by our model.

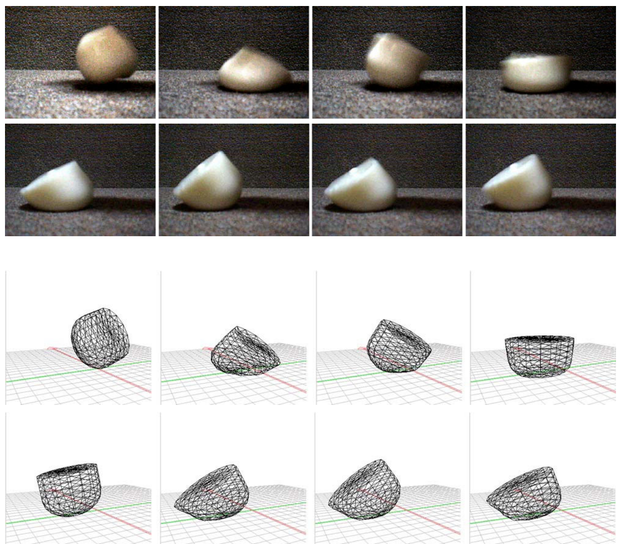

Figure 12: This figure shows the animation in different initial condition from Fig.11.

References

[1] Doug L.James and Dinesh K.Pai :" Accurate Real Time Deformable Objects ", SIGGRAPH 99, pp.65-72.

[2] Shinya Miyazaki, Shunsuke Yoshida, Takami Yasuda, and Shigeki Yokoi : " Proposal to Model Elastic Objects Based on Maintaining Local Shapes ", The Journal of The Institute of Electronics, Information and Communication Engineers, J82-A,7, pp.1148-1155, 1999.7.

[3] Vassilev, T., Spanlang, B., Chrysanthou Y.:" Efficient Cloth Model and Collision Detection for Dressing Virtual People", (in CD proc. GeTech Hong Kong), January 2001

[4] Gordon Collins, Adrian Hilton: "Models for character animation", Software Focus Volume 2, 44-51, Summer 2001

[5] James F.O'Brien, Jessica K. Hodgins: "Graphical Modeling and Animation of Brittle Fracture", SIGGRAPH 99, pp.137-146.

[6] Mero M. G., Susín A: "Deformable 3D Objects for a VR medical application"3es. Jornades de Recerca en Enginyeria Biomèdica. Lacroix D., Ginebra M.P. ed pp.264--269 2002

[7] Shinya MIYAZAKI, Takami YASUDA, Shigeki YOKOI, Junichiro TORIWAKI : "Modeling and Implementation of Elastic Object Manipulation Local Shapes", The Journal of The Institute of Electronics, Information and Communication Engineers, J79-A, 11, pp. 1919-1926, 1996.11

[8] G.Debunne, M.Dsbrun, M.Cani, and A.Barr. Dynamic real-time deformations using space and time adaptive sampling. In SIGGRAPH 2001 Conference Proceedings, 2001.

[9] J.Platt, A. Barr: "Constraint Methods for Flexible Models", Computer Graphics (SIGGRAPH '88 Proceedings), pp.279-288, August 1988

[10] Eitan Grinspun, Anil Hirani, Mathieu Desbrun and Peter Schröder : "Discrete Shells", Symposium on Computer Animation 2003