Physically Accurate Haptic Rendering of Elastic Object for a Haptic Glove

Chan-Su Lee

Abstract

This is a final report for the DIMACS grant of student-initiated project. I implemented Boundary Element Method(BEM) using Java. The system is tested with Rutgers Master II haptic glove. This report covers related background research, implementation of BEM and experiment results.

1 Introduction

Providing force feedback to a user (haptic feedback) has become an important topic of research in virtual environments and madecal simulation systems. There are many computational geometrical problems in computing haptics as in computer graphics. One of the most important issues is how to generate realistic deformation of elastic object and realistic contact forces during manipulation of virtual objects. In this project, I focus on real-time haptic force feedback generation using the haptic glove, the "Rutgers Master II-ND". I use Bounded Element Method(BEM) for the physical-based modeling of the elastic deformation.

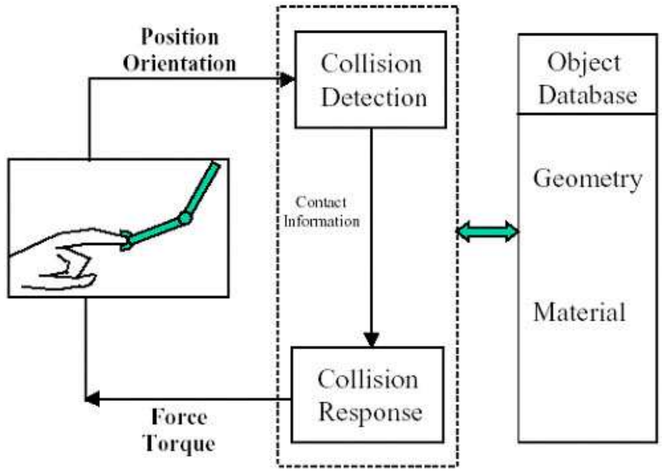


Figure 1: Haptic Interaction [1]

1.1 Problem Description

The problem is to **model realistic deformation of elastic object and calculate contact force for Rutgers Master II**. First, we need to find proper modeling for the realistic deformation which allows contact force calculation. Second, we have to consider what is the requirement for Rutgers Master II and how to take advantage of its features in the modeled haptic rendering. Finally, we have to consider real-time processing of the force generation. It is required to calculate and update the force fast.

1.2 Modeling for Haptic Rendering

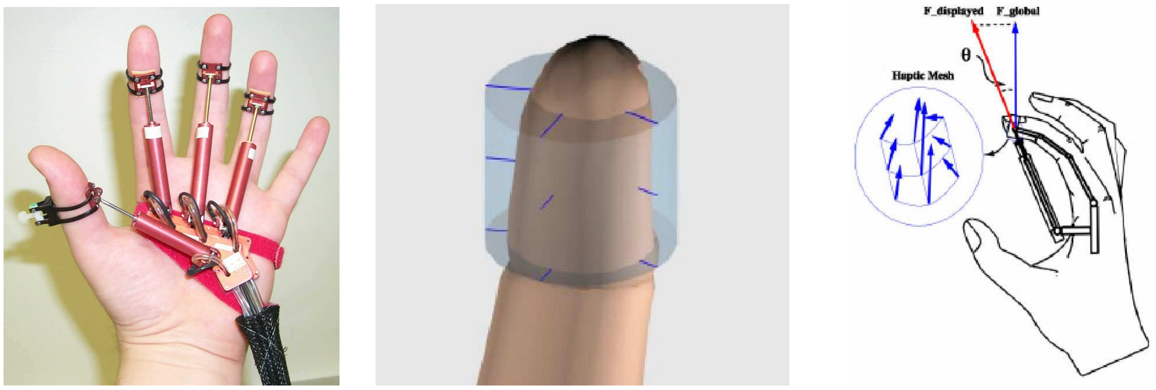
Haptic rendering is *the process of computing and generating forces in response to user interaction with virtual objects*[1]. It is necessary to calculate collision between virtual object and haptic device and reaction force as in figure 1. Haptic rendering can be divided as point-based, ray-based and 3D object based technique by the type of haptic interaction. In point-based haptic interaction, only the end-effector of the haptic device interacts with virtual objects. It is fast but difficult to generate interaction similar to real hand interaction to the world. It also has problem to generate torque. Ray-based interaction allows to calculate torque. 3D object-based interaction is desirable for many application, but this is computationally expensive than others. Deformable objects has been developed extensively in computer graphics. To apply them to haptic rendering, magnitude and direction of interaction forces have to be calculated.

There are deformation models based on geometry, finite approximation, and physically based model. To make more realistic hatpic feedback, physical based model approach is better. Physically-based models directly calculate interaction force without a separate models. Particle-based models, Finite element-based models, and meshless models are used[2].

1.3 Rutgers Master II

Rutgers Master II-New Design is haptic interface for dexterous hand interactions with virtual environments. Thumb, index, middle, and ring fingertips are connected to the pneumatic actuators. Actuators are connected to the L-shaped base, which is attached to the palm. Figure 2(a) shows its connection. Sensors can be updated 435 records/sec and each finger generates force up to 10N per finger with 12bit resolution[3]. It can control valves by 300 Hz and fingertips by 10Hz. It uses a virtual hand model to estimate joint angle of each finger.

This haptic device is different from other haptic interface in that it has four contact points which can be controlled separetely. Phantom^{TM}, one of the most famous commercial haptic devices, just support one contact points. In addition, this device usually uses 3D hand model to interact with virtual world. So, it really needs haptic rendering with a 3D hand model and objects.



(a) Rutgers Master II (b) Haptic Interaction mesh for a virtual fingertip (c) Force Mapping for RM-II

Figure 2: Rutgers Master II-ND & Haptic Rendering

1.4 Haptic Rendering for Rutgers Master II

One approach for 3D haptic rendering using haptic glove was developed based on haptic interaction mesh[4]. In order to model virtual hand in haptic interaction, it generalized haptic interaction point(HIP) based method to account for the shape of hand finger. It uses a set of points, called *haptic interaction mesh*, to model the haptic interaction of a force feedback glove. Figure 2(b) shows haptic interaction mesh(HIM) mapped on a virtual fingertip. The haptic interaction mesh used as a *ideal haptic interaction point*(IHIP). Forces are calculated based on Hooke's law and the displacement vector between HIP-IHIP. Force smoothing was done to reduce large variation of force in edges of polygnal surfaces by averaging normal vector similar to Phong shading in graphics. Computed force vector was mapped to the force displayed to the user's finger tip by considering the global interaction force as in figure 2(c). Surface deformation was also implemented by global morphing of hand motion and local deformation using contact of haptic interaction mesh.

This approach solves real-time haptic interaction with deformable or plastic objects. However, it did not consider the properties of material even though we can control stiffness of Hooks's law.

The haptic interaction mesh counts the shape of the virtual finger tip. But, it approximate the surface with several mesh lines. To solve such a problem, we need to use more physically correct solution of haptic force. We also need to consider contact surface of virtual hand. In addition to that, we have to support real time interaction with virtual environment. As a solution for these requirement, bounded element method(BEM) is one of the solution fit these requirement.

2 Boundary Elmemt Method

BEM is one of the methods to solve enginering problems using numerial method. It is also used for physical-based analysis of elasticity. Finite element method(FEM) divides a problem domain into elements and each element reproduces small region of the domain by enforcing continuity between the elements. However,FEM leads to very large number of finite elements as it discretize a whole body[5].

Instead of considering function which satisfies the boundary conditions, we can use functions which satisfy the governing equations and only approximately satisfies the boundary conditions[6]. BEM transforms the differential equations into equivalent integral ones. This makes discretization involve subdivision of the bounding surface of a body.

2.1 FEM vs BEM

BEM reduce the dimensitonality of the basic process by one: two-dimensional for three-dimensional problems. BEM does not calculate the whole inside structure in three-dimensional problems but the analysis generates only two-dimensional surface integral equation. It saves computation and is good for our application because we just contact object by the surface of 3D hand model.

BEM involves modelling the boundary geometry of the system. However, it is possible to calculate solutions of any subsequent interior points after solution is derived from the boundary information. The boundary integral equation is a statement of the exact solution to the given problem. Numerical integration is a much more stable and precise process than numerical differentiation.

2.2 Haptic Rendering using BEM

FEM and BEM techiniques are used for haptic rendering in the realistic medical simulation. Recently, BEM also began to be used for character animation for real-time deforamtion and haptic rendering by James and Pai[7]. They used precomputed Green's functions for global deformation of linear elastic objects and Capacitance Matrix Algorithms for fast low-rank updates. Their interaction is based on point-like contact and distributes forces using vertex presure masks. So, their system does not count surface contact in real 3D objects.

3 Implementation and Experiment

BEM is implemented for 3D haptic rendering using haptic glove device. There are three main steps to implement BEM for 3D elastic objects.

1. Discretize the boundary Γ into a set of N non-overlapping elements which represent the displacements and tractions.

2. Apply the integral equations at each of the n boundary nodes, and perform the resulting integrals over each boundary element in order to generate an undetermined system equations

3. Apply the boundary conditions of the desired boundary value problem, fixing n nodal values per direction

3.1 BEM implementation

It is required to describe the problem, such as boundary conditions and material properties, in a mathematical way. One of the way to describe surface or shape with reasonable number of data is to approximate using shape function. Serendipity shape fuctions and Lagrage shape functions are used in current implementation. One dimensional serendipity function in the Cartesian coordinates can be represented as in equation 1.

$$\begin{aligned} N_3(\zeta) &= 1 - \zeta^2 \\ N_n(\zeta) &= \tfrac{1}{2}(1 + \zeta_n \zeta) - \tfrac{1}{2} N_3 \, \zeta = 1, 2 \end{aligned} \tag{1}$$

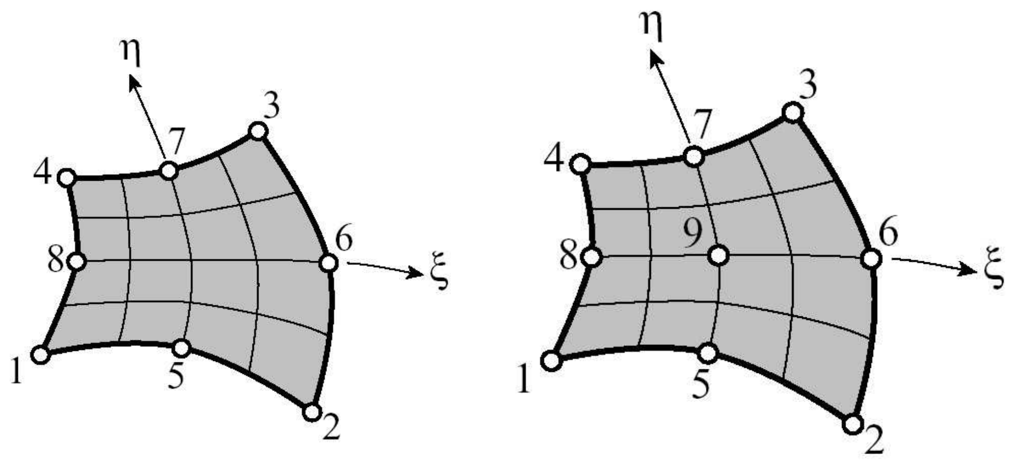
The shape functions have the following properties:

$$\begin{aligned} N_n(\zeta_n) &= 1 \\ N_n(\zeta_i) &= 0 \, for \, i \neq n \\ \sum N_n(\zeta) &= 1 \end{aligned}$$

Two dimentional Serendipity functions are extension of these properties with two intransic coordinates. Shape functions for the quadratic Lagrage element are expressed in equation 2. Figure 3 shows quadratic Serendipity element and quadratic Lagrange element.

$$\begin{aligned} L_n(i,j) &= A_{i1} A_{i2} A_{i3} B_{j1} B_{j2} B_{j3} \\ A_{il} &= \tfrac{\zeta - \zeta_l}{\zeta_i - \zeta_l} \, for \, i \neq l \\ A_{il} &= 1 \, for \, i = l \\ B_{jl} &= \tfrac{\eta - \eta_l}{\eta_j - \eta_l} \, for \, j \neq l \\ B_{jl} &= 1 \, for \, i = l \end{aligned} \tag{2}$$

Physical quantites of a model are specified at each nodel point and interpolated using shape function. Differentiation of surface geometry and integration over elements are also implemented based on shape function. In the implementation of boundary element method, the integration



(a) Quadratic Serendipity element (b) Quadratic Lagrange element

Figure 3: Shape Function Elements

plays important roles because we have to be able to integrate over quite complex surface. For more accurate integration, Gauss quadrature with a variable number of integration points are used. The number of integration points and weighting parameter kept in lookup table. The basic funtion for two-dimentional element is shown in equation 3.

$$I = \int_{-1}^{+1} \int_{-1}^{+1} f(\zeta, \eta) d\zeta d\eta \approx \sum_{i=1}^{N} \sum_{j=1}^{M} W_i W_j f(\zeta_i, \eta_j) \tag{3}$$

BEM can be used for analysis of homogeneous domains with linear elastic material as we can get fundamental solution in that case. Inhomogeneous materials can be analysed by the multi-region approach. In this project, homogeneous domains with linear elastic material are considered.

Fundamental solution for static elastic problems was first worked out by Kelvin and can be calculated for unit load in each direction. The solutions for the displacements in the x,y and z directions due to a unit load in the x-direction can be written as

$$\begin{aligned} U_{xx}(P,Q) &= \frac{C}{r}\left(C_1 + \left(\frac{r_x}{r}\right)^2\right) \\ U_{xy}(P,Q) &= \frac{C}{r}\left(\frac{r_x}{r}\right)\left(\frac{r_y}{r}\right) \\ U_{xz}(P,Q) &= \frac{C}{r}\left(\frac{r_x}{r}\right)\left(\frac{r_z}{r}\right) \end{aligned}$$

$$where\,C \;=\; 1/(16\pi G(1-v)),\; C_1 = 3-4v$$

The fundamental solutions for tractions due to a unit load at P in the x-direction are as follows. Simliar solutions of displacements and tractions can be found for y and z-directional unit load.

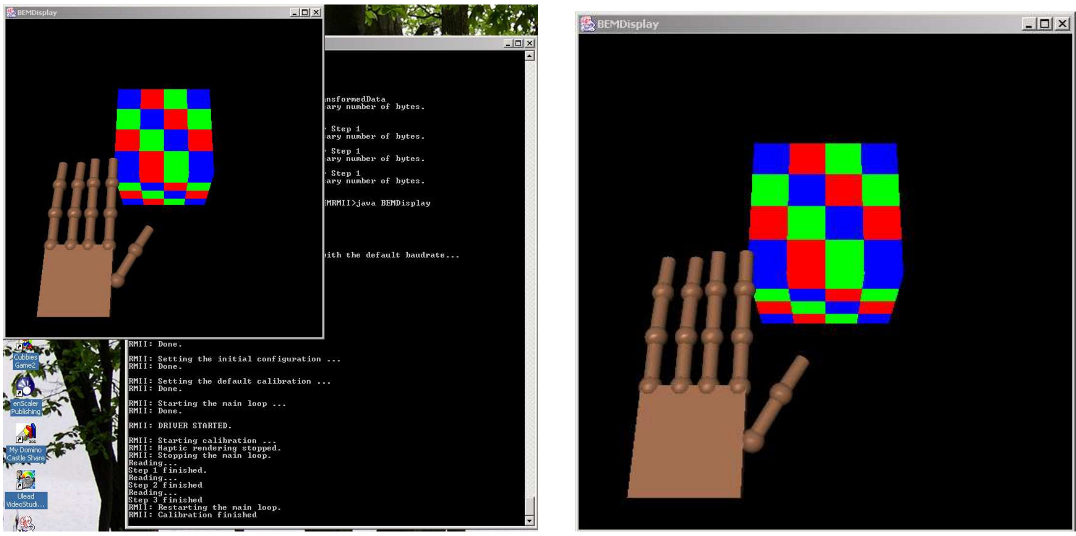
$$\begin{aligned}
T_{xx}(P,Q) &= -\frac{C_2}{r^2}\left(C_3 + 3\left(\frac{r_x}{r}\right)^2\right)cos\theta \\
T_{xy}(P,Q) &= -\frac{C_2}{r^2}\left(3\left(\frac{r_x}{r}\right)\left(\frac{r_y}{r}\right)cos\theta - C_3\left(n_y\left(\frac{r_x}{r}\right) - n_x\left(\frac{r_y}{r}\right)\right)\right) \\
T_{xz}(P,Q) &= -\frac{C_2}{r^2}\left(3\left(\frac{r_x}{r}\right)\left(\frac{r_z}{r}\right)cos\theta - C_3\left(n_z\left(\frac{r_x}{r}\right) - n_x\left(\frac{r_z}{r}\right)\right)\right) \\
where\,C_2 &= -1/(8\pi(1-v)),\; C_3 = 1-2v \\
cos\theta &= (fracr_x r)\,n_x + \left(\frac{r_y}{r}\right)n_y + \left(\frac{r_z}{r}\right)n_z
\end{aligned}$$

Based on the fundamental solution, numerical integration is done using shape function. Integral equation for the analysis is changed to a sum of integrations of Kernel shape function products over elements. By assembly procedure, each element contributions are put together into the global coefficient matrix, which has all known variables on the right side and all unknown variables on the left side of the equation. The integral equations are solved by solving the linear system equation using Gauss elimination. By postprocessing, we get solution for the inside of the boundary from the boundary solution.

3.2 Experiments with RM II

The whole BEM algorithm is implemented by Java. Colt version 1.0.3 is used for matrix multiplication and solving the linear system equation. Colt is an open source libraries for high performance scientific and tecnnical computing in Java. More than 3,000 lines of code is implemented purely for the BEM algorithm.

The display of the BEM elements and hand model is implemented using Java 3D. The Rutgers Master II is connected to the system using JNI. It allows to get angle data of each joint. Hand



(a) Screen image of the running program (b) Initial Boundary Condition

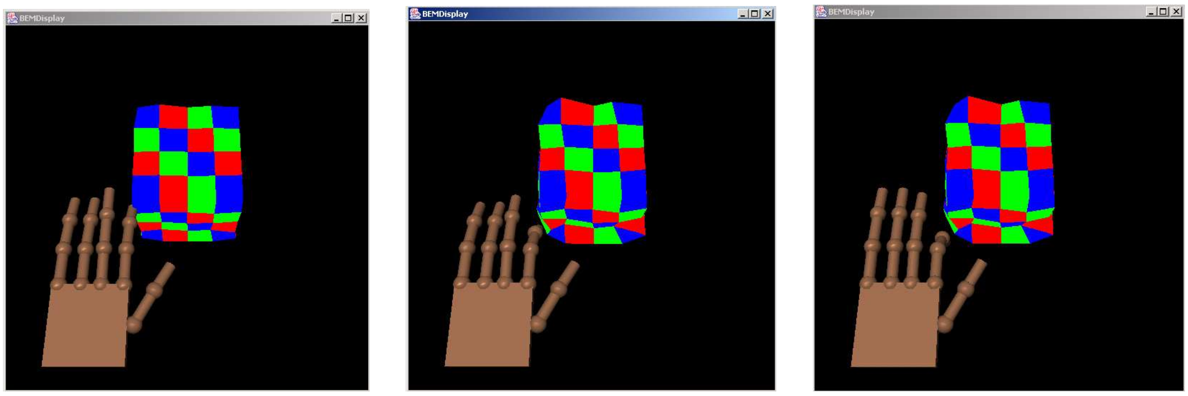
Figure 4: Experiemtn Setup

model changes joint angle based on the sensing data of RM II. Current implementation does not check exact collision detection to manipulate model using hand model. Instead, it just checks angles of index finger and calculate its location. The calculated location is used to change of boundary condition. Based on the changed boundary condition, the whole system equation is solved again. It takes a little longer time. So, new system equation is solved every 5 frames.

Figure 4 shows initial state of the model. Figure 5 shows screen shot of the experiments. It shows different deformation in different boundary condition.

4 Conclusion and Future Work

BEM is implemented using Java and Java3D. It is connected RMII haptic device. This system does not give real haptic feedback but it just uses angle data to change the boundary condition. As the whole system solution is calculated whenever the boundary condition is changed, it is very slow. It can be improved by using partial update of the system matrix instead of the whole matrix.



(a) Boundary condition 1 (b) Boundary condition 2 (c) Boundary condition 3

Figure 5: Test results

References

[1] K. Salisbury, D. Brock, N. Swarup, and C. Zilles. Haptic rendering: Programming touch interaction with virtual world. In *Proceeding of 1995 ACM Symposium on Interactive 3D Graphics*.

[2] K. Stanney, editor. *Handbook of Virtual Environments:Design, Implementation, and Applications(Human Factors and Ergonomics)*. Lawrence Erlbaum Assoc, February 2002.

[3] M. Bouzit, G. Burdea, G. Popescu, and R. Boian. The rutgers master ii–new design force-feedback glove. *IEEE/ASME Transactions on Mechatronics*, 7(2), June 2002.

[4] G. Burdea M. Bouzit Popescu, V. Virtual reality simulation modeling for a haptic glove. In *Computer Animation'99 Conference*, pages 195–200, May 1999.

[5] P.K. Banerjee and R. Butterfield. *Boundary Element Methods in Engineering Science*. McGRAW-HILL, 1981.

[6] C.A. Brebbia, J.C.F. Telles, and L.C. Wrobel. *Boundary Element Techinques: Theory and Application in Engineering*. Springer-Verlag, 1984.

[7] D.L. James and D.K. Pai. A unified treatment of elastostatic contact simulation for real time haptics. *Haptics-e:The Electronic Journal of Haptic Research*, 2(1), 2001.