

IROS 2007
Oct. 29 – Nov.2, 2007
San Diego, U.S.A

Modeling, Identification and Control of Deformable Soft Objects

Organizer:

Shinichi Hirai
Department of Robotics,
Ritsumeikan University

Co-organizer:

Penglin Zhang
School of Remote Sensing and
Information Engineering,
Wuhan University

Full Day Workshop

November 2, 2007
8:30 am – 5:00 pm
FW-4

Modeling, Identification, and Control of Deformable Soft Objects

Full-Day Workshop at
2007 IEEE/RSJ International Conference on Intelligent Robots and Systems
(IROS 2007)
San Diego, California USA
Friday, November 2, 2007

Organizers

Shinichi Hirai

Professor of Department of Robotics
Ritsumeikan University,
Noji-higashi 1-1-1,
Room 3437-0, East Wing 4F,
Kusatsu, Shiga 525-8577
+81-77-561-2879
+81-77-561-2665 (Fax)
Hirai@se.ritsumeai.ac.jp
<http://www.ritsumeai.ac.jp/~hirai/>

Penglin Zhang

Associate Professor of School of
Remote Sensing and Information
Engineering, Wuhan University
PD of Ritsumeikan University,
Room 245, East Wing 4F.
Kusatsu, Shiga 525-8577
+81-77-561-2879
+81-77-561-2665 (Fax)
zpl@se.ritsumeai.ac.jp

Summary

Researches on deformable soft objects such as biological tissue, food dough, thread, and wire harness are now one of emerging issues in virtual reality, computer vision, medical engineering, and robotics. We have to tackle many topics including geometric and mechanical modeling of deformable soft objects, their model identification, and control of their deformation to treat the deformable soft objects in engineering. This workshop focuses on the current researches on modeling, identification, and control of deformable soft objects. Topics are 1) geometric/mechanical modeling of deformable soft objects, in robotics, automation science, and medical engineering, 2) identification of deformation model parameters, 3) realtime simulation of object deformation, 4) control of object deformation in robotics and automation, 5) manipulation and handling of deformable soft objects, and other related issues.

Modeling, Identification, and Control of Deformable Soft Objects

Table of Contents/Schedule

			Pages
Welcome & Introduction			
8:30 AM	Coffee and Pastries		
8:45 AM	Shinichi Hirai		i
	Contents/Schedule		ii
Modeling of Soft Material Objects			1
9:00AM	Hiroyuki Fujioka	Tokyo Denki University	2-7
9:35AM	Penglin Zhang	Ritsumeikan University	8-13
10:10AM	Coffee Break		
Modeling of Deformable Linear Objects			14
10:25AM	Hidefumi Wakamatsu	Osaka University	15-24
11:00AM	Hans Fuhan Shi	Simon Fraser University	25-32
11:35AM	Lunch Break		
Identification of Deformation			33
13:00PM	Hiroshi Noborio	Osaka Electro-Communication University	34-90
13:35PM	Yasuyo Kita	National Institute of Advanced Industrial Science and Technology	91-95
14:10PM	Coffee Break		
Control of Deformable Soft Objects			96
14:25PM	Satoshi Yamaguchi	Ritsumeikan University	97-104
15:00PM	Mizuho Shibata	Ritsumeikan University	105-113
15:35PM	Coffee Break		
Conclusion			
15:50PM	Panel Discussion		
17:00PM	End		

Modeling of Soft Material Objects

Presentation 1: *Contour and Shape Modeling of Wet Material Objects using Periodic and Closed Spline Surfaces*

Pages: 2–7

Presentation 2: *A Method for Non-rigid 3D Deformation Fields Measurement: Application to Human Calf MR Volumetric Images*

Pages: 8–13

Contour and Shape Modeling of Wet Material Objects using Periodic and Closed Spline Surfaces

Hiroyuki Fujioka and Hiroyuki Kano

Abstract—In this paper, we present a synthesizing scheme for modeling the contour and shape of wet material objects using optimal periodic and closed surfaces. The surfaces are constructed by employing normalized uniform B-splines as the basis functions. A concise representation for the optimal surfaces is derived, which has the additional merit of lending itself to the development of computational procedures in a straightforward manner. Moreover, the asymptotical and statistical properties of optimal surfaces are presented. The results are applied to the problem of modeling contour and shape of wet material objects with deforming motion, and the effectiveness is examined by numerical and experimental studies.

I. INTRODUCTION

Wet material objects such as jellyfish, red blood cell and amoeba, etc., move with various deformation motions. One of important issues in their studies is to analyze and understand such motions from the observational data, e.g. some image frames in a movie file. Then the contour and shape modeling may play the key role.

The problem of modeling the contour and shape of deformable objects has been studied in the field of image processings. In the works, spline functions have been used frequently [1], and various techniques, e.g. snakes and active contour model [2], have been developed. However, the most approaches have focused their attention on the problem of modeling the contour of objects at some time instant, and yield the difficulties to analyze and understand a whole motion of the moving deformable objects, i.e. wet material objects. Thus, a scheme of modeling the contour and shape of the deformable objects is required.

An approach to the development of such a scheme would be to design the periodic and closed surfaces by interpolating or smoothing a set of given discrete contour/shape data in some 3-dimensional space. In particular, employing the periodic surfaces and a 3-D space composed of the 2-dimensional image plane and the time axis, we may model the contour dynamically, which may be useful to analyze and understand a whole motion. This idea is similar to the one for the spline-based solid modeling of the human organs from a set of tomogram data obtained by the magnetic range imaging (MRI) [3], etc. It is recognized, however, that the interpolation often results in an oscillating surfaces, and hence inappropriate in such cases where the image data may include some noises. On the other hand, the approximation

by smoothing splines is stable numerically and yields feasible approximation results.

In this paper, we develop a synthesizing scheme for modeling the contour and shape of wet material objects based on the design method of optimal periodic and closed smoothing surfaces. The surfaces are constructed by employing normalized uniform B-splines as the basis functions. We first develop the method for designing optimal periodic and closed smoothing spline surfaces. Then, assuming that a number of data is given by sampling some surfaces $f(s, t)$ with noises, we analyze statistical properties of optimal smoothing splines and derive an expression of the splines as a functional of $f(s, t)$ when the number tends to infinity. Finally, the results are applied to model contour and shape of wet material objects such as red blood cell and jellyfish. The effectiveness are verified by numerical and experimental studies. Also, we show that the proposed method is helpful for analyzing and understanding the motions of wet material objects.

For designing surfaces $x(s, t)$, we employ normalized, uniform B-spline function $B_k(t)$ of degree k as the basis functions,

$$x(s, t) = \sum_{i=-k}^{m_1-1} \sum_{j=-k}^{m_2-1} \tau_{i,j} B_k(\alpha(s - s_i)) B_k(\beta(t - t_j)), \quad (1)$$

where $\alpha, \beta (> 0)$ are constants, $m_1, m_2 (> 2)$ are integers, and s_i 's, t_j 's are equally spaced knot points with $s_{i+1} - s_i = \frac{1}{\alpha}$, $t_{j+1} - t_j = \frac{1}{\beta}$.

In the sequel, we briefly describe the normalized, uniform B-spline functions: $B_k(t)$ is defined by

$$B_k(t) = \begin{cases} N_{k-j,k}(t-j) & j \leq t < j+1 \quad j=0, \dots, k \\ 0 & t < 0, k+1 \leq t. \end{cases} \quad (2)$$

Here the basis elements $N_{j,k}(t)$ ($j=0, 1, \dots, k$) are obtained recursively by the following algorithm [4]. Let $N_{0,0}(t) \equiv 1$ and, for $i=1, 2, \dots, k$, compute

$$\begin{cases} N_{0,i}(t) &= \frac{1-t}{i} N_{0,i-1}(t) \\ N_{j,i}(t) &= \frac{i-j+t}{i} N_{j-1,i-1}(t) \\ &+ \frac{1+j-t}{i} N_{j,i-1}(t), j=1, \dots, i-1 \\ N_{i,i}(t) &= \frac{t}{i} N_{i-1,i-1}(t). \end{cases} \quad (3)$$

Thus, $B_k(t)$ is a piece-wise polynomial of degree k with integer knot points and is $k-1$ times continuously differentiable. It is noted that $B_k(t)$ for $k=0, 1, 2, \dots$ is normalized in the sense of $\sum_{j=0}^k N_{j,k}(t) = 1$, $0 \leq t \leq 1$, and this yields $\int_{-\infty}^{\infty} B_k(t) dt = \int_0^{k+1} B_k(t) dt = 1$.

H. Fujioka and H. Kano are with the Dept. of Information Sciences, Tokyo Denki University, Saitama 350-0394, JAPAN. email: fujioka@j.dendai.ac.jp, kano@j.dendai.ac.jp, tel.: +81-49-296-2911, fax.: +81-49-296-6403

For the sake of later reference, we show here the function $B_3(t)$,

$$B_3(t) = \begin{cases} \frac{1}{6}t^3 & 0 \leq t < 1 \\ \frac{1}{6}(-3t^3 + 12t^2 - 12t + 4) & 1 \leq t < 2 \\ \frac{1}{6}(3t^3 - 24t^2 + 60t - 44) & 2 \leq t < 3 \\ \frac{1}{6}(4-t)^3 & 3 \leq t < 4 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

II. OPTIMAL PERIODIC AND CLOSED SMOOTHING SPLINE SURFACE

In this section, we develop the method for designing optimal periodic and closed smoothing splines. For simplicity, we restrict ourselves to the case of $k = 3$ in the sequel.

Equation (1) in the case of $k = 3$ is written as

$$x(s, t) = \sum_{i=-3}^{m_1-1} \sum_{j=-3}^{m_2-1} \tau_{i,j} B_3(\alpha(s-s_i)) B_3(\beta(t-t_j)). \quad (5)$$

Then, choosing appropriate weighting coefficient $\tau_{i,j}$ called 'control point', $x(s, t)$ can represent an arbitrary spline surface on the rectangular domain $\mathcal{S} = [s_0, s_{m_1}] \times [t_0, t_{m_2}] \subset \mathbf{R}^2$. Now suppose that a set of spatial data

$$\mathcal{D} = \{(u_i, v_j; d_{ij}) : (u_i, v_j) \in \mathcal{S}, d_{ij} \in \mathbf{R}, i = 1, 2, \dots, N_1, j = 1, 2, \dots, N_2\} \quad (6)$$

is given, and let $\tau \in \mathbf{R}^{M_1 \times M_2}$ be the weight matrix

$$\tau = [\tau_{i,j}]_{i,j=-3}^{i=m_1-1, j=m_2-1}, \quad (7)$$

where $M_1 = m_1 + 3$ and $M_2 = m_2 + 3$.

Then, a problem of designing optimal periodic/closed smoothing spline surfaces is to find a surface $x(s, t)$, or equivalently a matrix $\tau \in \mathbf{R}^{M_1 \times M_2}$, minimizing a cost function

$$J(\tau) = \lambda \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (\nabla^2 x(s, t))^2 ds dt + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_{ij} (x(u_i, v_j) - d_{ij})^2, \quad (8)$$

subject to continuity constraints

$$\frac{\partial^l}{\partial t^l} x(s, t_0) = \frac{\partial^l}{\partial t^l} x(s, t_{m_2}), \quad s \in [s_0, s_{m_1}] \quad (9)$$

or/and

$$\frac{\partial^l}{\partial s^l} x(s_0, t) = \frac{\partial^l}{\partial s^l} x(s_{m_1}, t), \quad t \in [t_0, t_{m_2}] \quad (10)$$

for $l = 0, 1, 2$. Here, $\lambda (> 0)$ is a smoothing parameter, and w_{ij} ($0 \leq w_{ij} \leq 1$) denotes weights for approximation errors.

These problems can be solved as follows: First, we express the right hand side of (8) in terms of τ . Let $b_1(t) \in \mathbf{R}^{M_1}$ and $b_2(t) \in \mathbf{R}^{M_2}$ be

$$b_1(s) = [B_3(\alpha(s-s_{-3})) \ B_3(\alpha(s-s_{-2})) \ \dots \ B_3(\alpha(s-s_{m_1-1}))]^T \quad (11)$$

$$b_2(t) = [B_3(\beta(t-t_{-3})) \ B_3(\beta(t-t_{-2})) \ \dots \ B_3(\beta(t-t_{m_2-1}))]^T. \quad (12)$$

Then with

$$\hat{\tau} = \text{vec } \tau \ (\in \mathbf{R}^{M_1 M_2}), \quad (13)$$

$x(s, t)$ in (5) can be written as

$$x(s, t) = (b_2(t) \otimes b_1(s))^T \hat{\tau}, \quad (14)$$

and the cost function in (8) is obtained in terms of $\hat{\tau}$ as

$$J(\hat{\tau}) = \lambda \hat{\tau}^T Q \hat{\tau} + ((B_2 \otimes B_1)^T \hat{\tau} - d)^T W ((B_2 \otimes B_1)^T \hat{\tau} - d). \quad (15)$$

Here, matrices $B_1 \in \mathbf{R}^{M_1 \times N_1}$ and $B_2 \in \mathbf{R}^{M_2 \times N_2}$ are defined by

$$B_1 = [b_1(u_1) \ b_1(u_2) \ \dots \ b_1(u_{N_1})], \\ B_2 = [b_2(v_1) \ b_2(v_2) \ \dots \ b_2(v_{N_2})], \quad (16)$$

and $W \in \mathbf{R}^{N_1 N_2 \times N_1 N_2}$ and $d \in \mathbf{R}^{N_1 N_2}$ by

$$W = \text{diag}\{w_{11}, w_{21}, \dots, w_{N_1 1}, \dots, \\ w_{1N_2}, w_{2N_2}, \dots, w_{N_1 N_2}\} \\ d = [d_{11}, d_{21}, \dots, d_{N_1 1}, \dots, \\ d_{1N_2}, d_{2N_2}, \dots, d_{N_1 N_2}]^T. \quad (17)$$

In (15), $Q \in \mathbf{R}^{M_1 M_2 \times M_1 M_2}$ is a Gramian defined by

$$Q = Q_2^{(00)} \otimes Q_1^{(22)} + Q_2^{(02)} \otimes (Q_1^{(02)})^T \\ + (Q_2^{(02)})^T \otimes Q_1^{(02)} + Q_2^{(22)} \otimes Q_1^{(00)}, \quad (18)$$

where $Q_l^{(ij)} \in \mathbf{R}^{M_l \times M_l}$ ($l = 1, 2; i, j = 0, 1, 2$) are given by

$$Q_l^{(ij)} = \int_{I_l} \frac{d^i b_l(t)}{dt^i} \frac{d^j b_l^T(t)}{dt^j} dt \quad (19)$$

with $I_1 = [s_0, s_{m_1}]$ and $I_2 = [t_0, t_{m_2}]$. Each matrix Q_l in (18) can be computed a priori (i.e. regardless of the data d_{ij}) when the relevant parameters such as m_1 and m_2 are specified. Thus, the matrix Q is computed explicitly. Also, it can be shown that the matrix Q is singular (see [5]).

Next we express the constraints (9) and (10) in terms of τ . Letting $\tau_i^c \in \mathbf{R}^{M_1}$, $i = -3, -2, \dots, m_2 - 1$, be the i -th column vector of the matrix τ in (7), i.e.

$$\tau = [\tau_{-3}^c \ \tau_{-2}^c \ \dots \ \tau_{m_2-1}^c], \quad (20)$$

we can show that (9) for $l = 0, 1, 2$ is written respectively as

$$\frac{1}{6} \tau_{-3}^c + \frac{4}{6} \tau_{-2}^c + \frac{1}{6} \tau_{-1}^c = \frac{1}{6} \tau_{m_2-3}^c + \frac{4}{6} \tau_{m_2-2}^c + \frac{1}{6} \tau_{m_2-1}^c \\ \tau_{-3}^c - \tau_{-1}^c = \tau_{m_2-3}^c - \tau_{m_2-1}^c$$

$$\tau_{-3}^c - 2\tau_{-2}^c + \tau_{-1}^c = \tau_{m_2-3}^c - 2\tau_{m_2-2}^c + \tau_{m_2-1}^c,$$

yielding $\tau_{-3}^c = \tau_{m_2-3}^c$, $\tau_{-2}^c = \tau_{m_2-2}^c$, $\tau_{-1}^c = \tau_{m_2-1}^c$. Then, letting $G_l \in \mathbf{R}^{M_1 \times M_1 M_2}$, $l = 1, 2$ be the matrix defined by

$$G_l = [I_{3M_1 \times 3M_1} \ 0_{3M_1 \times (M_1 M_2 - 6M_1)} \ -I_{3M_1 \times 3M_1}], \quad (21)$$

the constraint in (9) is written as a linear constraint,

$$G_1 \hat{\tau} = 0. \quad (22)$$

Similarly, the constraint in (10) is obtained as

$$G_2 E \hat{\tau} = 0 \quad (23)$$

with a computation matrix $E \in \mathbf{R}^{M_1 M_2 \times M_1 M_2}$, i.e. $\text{vec } \tau^T = E \hat{\tau}$. Thus the constraints for periodic and closed cases can be written as

$$\bar{G}_l \hat{\tau} = 0, \quad l = 1, 2, 3, \quad (24)$$

where $\bar{G}_1 = G_1$ and $\bar{G}_2 = G_2 E$ are for periodic cases, and $\bar{G}_3 = [\bar{G}_1^T \bar{G}_2^T]^T \in \mathbf{R}^{3M_3 \times M_1 M_2}$ ($3M_3 = 3M_1 + 3M_2$) for closed case.

Minimizing the cost function subject to the constraints (9) or/and (10) is now a straightforward task. For the cost function in (8), i.e. (15), we form the following Lagrangian function,

$$L(\hat{\tau}, \mu) = \lambda \hat{\tau}^T Q \hat{\tau} + (B^T \hat{\tau} - d)^T W (B^T \hat{\tau} - d) + \mu_l^T \bar{G}_l \hat{\tau}, \quad (25)$$

where $\mu \in \mathbf{R}^{3M_l}$, $l = 1$ or $l = 2$ is for the periodic case, and $l = 3$ for the closed case. Then, by taking derivatives with respect to $\hat{\tau}$ and μ , we get

$$\begin{aligned} \lambda Q \hat{\tau} + B W B^T \hat{\tau} + \frac{1}{2} \bar{G}_l^T \mu_l &= B W d \\ \bar{G}_l \hat{\tau} &= 0, \end{aligned} \quad (26)$$

or

$$\begin{bmatrix} \lambda Q + B W B^T & \bar{G}_l^T \\ \bar{G}_l & 0_{3M_l \times 3M_l} \end{bmatrix} \begin{bmatrix} \hat{\tau} \\ \frac{1}{2} \mu_l \end{bmatrix} = \begin{bmatrix} B W d \\ 0_{3M_l} \end{bmatrix}. \quad (27)$$

It can be shown that this equation is consistent, i.e.

$$\begin{aligned} \text{rank} \begin{bmatrix} \lambda Q + B W B^T & \bar{G}_l^T & B W d \\ \bar{G}_l & 0_{3M_l \times 3M_l} & 0_{3M_l} \end{bmatrix} \\ = \text{rank} \begin{bmatrix} \lambda Q + B W B^T & \bar{G}_l^T \\ \bar{G}_l & 0_{3M_l \times 3M_l} \end{bmatrix}. \end{aligned} \quad (28)$$

Here, in (28), if $\lambda Q + B W B^T > 0$ and \bar{G}_l is of row full rank, then the coefficient matrix is nonsingular, and the solution exists uniquely. If it is singular, we employ the minimum norm solution, yielding unique $\hat{\tau}$.

III. ASYMPTOTICAL AND STATISTICAL PROPERTIES

In this section, we present asymptotical and statistical properties of optimal periodic and closed smoothing surfaces when the number of sampled data tends to infinity and the data contains noises.

Now we assume that the data d_{ij} in (6) for constructing smoothing surfaces is obtained by sampling a function $f(s, t)$ which is assumed to be continuous in both variables in $[s_0, s_{m_1}] \times [t_0, t_{m_2}]$. In order to analyze asymptotic properties of spline surfaces as the number of data points N_1, N_2 increases, we employ the following cost function instead of (8),

$$\begin{aligned} J_{N_1, N_2}(\tau) &= \lambda \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (\nabla^2 x(s, t))^2 ds dt \\ &+ \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (x(u_i, v_j) - f(u_i, v_j))^2. \end{aligned} \quad (29)$$

When the data d_{ij} is obtained by sampling the function $f(s, t)$ with additive noises

$$d_{ij} = f(s_i, t_j) + \varepsilon_{ij}, \quad i = 1, 2, \dots, N_1, \quad j = 1, 2, \dots, N_2, \quad (30)$$

we employ a cost function

$$\begin{aligned} J_{N_1, N_2}^{\varepsilon}(\tau) &= \lambda \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (\nabla^2 x(s, t))^2 ds dt \\ &+ \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (x(u_i, v_j) - f(u_i, v_j) - \varepsilon_{ij})^2. \end{aligned} \quad (31)$$

Here, the noises ε_{ij} are assumed to be zero-mean and white, i.e. $E\{\varepsilon_{ij}\} = 0 \quad \forall i, j$, $E\{\varepsilon_{ij} \varepsilon_{kl}\} = \sigma^2$ for $i = j = k = l$, and $E\{\varepsilon_{ij} \varepsilon_{kl}\} = 0$ otherwise. Moreover, in order to analyze convergence properties of solutions as $N_1, N_2 \rightarrow \infty$, we introduce the following cost function,

$$\begin{aligned} J_c(\tau) &= \lambda \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (\nabla^2 x(s, t))^2 ds dt \\ &+ \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (x(s, t) - f(s, t))^2 ds dt. \end{aligned} \quad (32)$$

Using the results in Section II, the optimal solutions τ_{N_1, N_2} and $\tau_{N_1, N_2}^{\varepsilon}$ minimizing the cost functions in (29) and (31) subject to the constraints in (9) or/and (10) are obtained respectively as solutions of

$$\begin{bmatrix} \mathcal{A}_{N_1, N_2} & \bar{G}_l^T \\ \bar{G}_l & 0_{3M_l \times 3M_l} \end{bmatrix} \begin{bmatrix} \hat{\tau} \\ \frac{1}{2} \mu \end{bmatrix} = \begin{bmatrix} \frac{1}{N_1 N_2} B f \\ 0_{3M_l} \end{bmatrix}, \quad (33)$$

and

$$\begin{bmatrix} \mathcal{A}_{N_1, N_2} & \bar{G}_l^T \\ \bar{G}_l & 0_{3M_l \times 3M_l} \end{bmatrix} \begin{bmatrix} \hat{\tau} \\ \frac{1}{2} \mu \end{bmatrix} = \begin{bmatrix} \frac{1}{N_1 N_2} B(f + \varepsilon) \\ 0_{3M_l} \end{bmatrix}, \quad (34)$$

with $l = 1, 2, 3$, and $\mathcal{A}_{N_1, N_2} \in \mathbf{R}^{M_1 M_2 \times M_1 M_2}$ is given by

$$\mathcal{A}_{N_1, N_2} = \lambda Q + \frac{1}{N_1 N_2} B B^T, \quad (35)$$

and the vectors $f \in \mathbf{R}^{N_1 N_2}$ and $\varepsilon \in \mathbf{R}^{N_1 N_2}$ are defined by

$$\begin{aligned} f &= [f(u_1, v_1), f(u_2, v_1), \dots, f(u_{N_1}, v_1), \dots, \\ &f(u_1, v_{N_2}), f(u_2, v_{N_2}), \dots, f(u_{N_1}, v_{N_2})]^T, \\ \varepsilon &= [\varepsilon_{11}, \varepsilon_{21}, \dots, \varepsilon_{N_1 1}, \dots, \\ &\varepsilon_{1N_2}, \varepsilon_{2N_2}, \dots, \varepsilon_{N_1 N_2}]^T. \end{aligned}$$

On the other hand, we can show that the optimal solution τ_c for the cost function in (32) is obtained as the solution of

$$\begin{bmatrix} \lambda Q + Q_2^{(00)} \otimes Q_1^{(00)} & \bar{G}_l^T \\ \bar{G}_l & 0_{3M_l \times 3M_l} \end{bmatrix} \begin{bmatrix} \hat{\tau} \\ \frac{1}{2} \mu \end{bmatrix} = \begin{bmatrix} \phi \\ 0_{3M_l} \end{bmatrix}, \quad (36)$$

where $\phi \in \mathbf{R}^{M_1 M_2}$ is defined by

$$\phi = \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} f(s, t) (b_2(t) \otimes b_1(s)) ds dt. \quad (37)$$

Note that the coefficient matrix \mathcal{A}_{N_1, N_2} in (33) and (34) may be singular since the matrix Q is singular, in which case we use the minimum norm solution, yielding a unique solution τ_{N_1, N_2} , $\tau_{N_1, N_2}^{\varepsilon}$. On the other hand, (36) always has a unique solution τ_c since it holds that the matrix $Q_2^{(00)} \otimes Q_1^{(00)}$ is positive-definite.

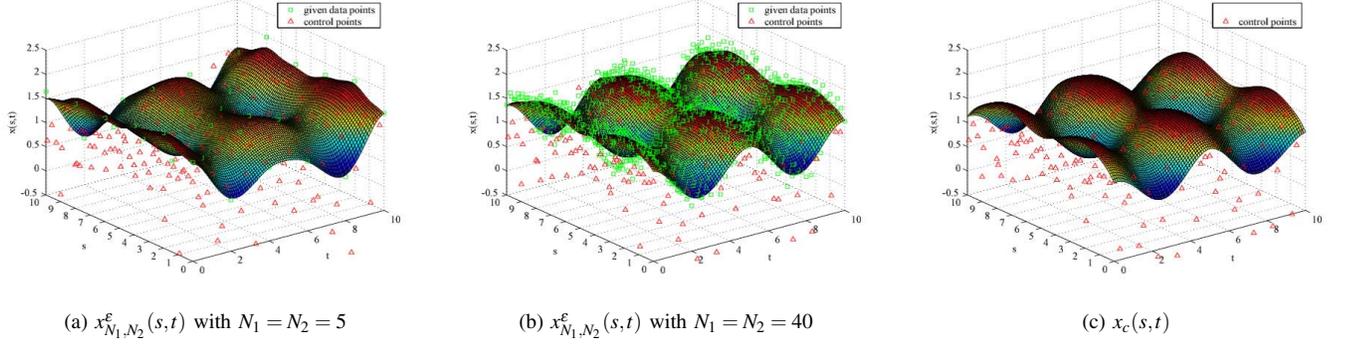


Fig. 1. Optimal periodic smoothing surfaces.

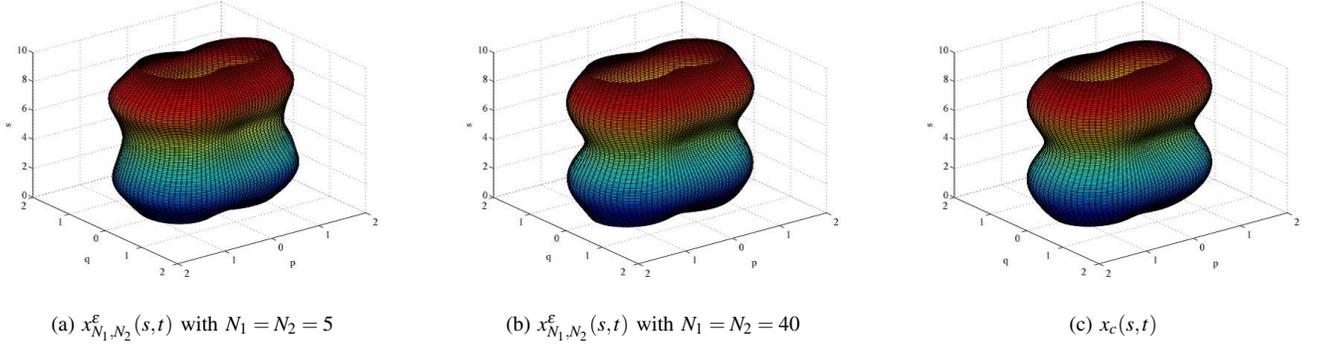


Fig. 2. Periodic surfaces $x(s, t)$ represented in the $o-pqs$ space.

Now we introduce the following assumption:

Assumption 1: The sample points (u_i, v_j) , $i = 1, \dots, N_1$, $j = 1, \dots, N_2$, are such that

$$\lim_{N_1, N_2 \rightarrow \infty} \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} g(u_i, v_j) = \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} g(s, t) ds dt$$

for every continuous function $g(s, t)$ in $[s_0, s_{m_1}] \times [t_0, t_{m_2}]$.

Then we obtain the following result.

Theorem 1: Assume that the condition of Assumption 1 holds, and let τ_{N_1, N_2} , τ_{N_1, N_2}^e be the minimum norm solutions of (33) and (34) respectively. Then,

- (i) τ_{N_1, N_2} converges to τ_c as $N_1, N_2 \rightarrow \infty$.
- (ii) $E\{\tau_{N_1, N_2}^e\} = \tau_{N_1, N_2}$ and τ_{N_1, N_2}^e converges to τ_c as $N_1, N_2 \rightarrow \infty$ in the mean squares sense.

IV. CONTOUR AND SHAPE MODELING

We apply the method for designing optimal periodic and closed smoothing splines to the problem of contour and shape modeling. We set $k = 3$ and $\alpha = \beta = 1$ in below.

A. Numerical Study

For the periodic case, we generate the data \mathcal{D} by sampling the following periodic surface,

$$f(s, t) = \sqrt{\cos \theta(t) + \sqrt{b^4(s) - \sin^2 \theta(t)}} \quad (38)$$

with $\theta(t) = \frac{36\pi}{180}t$ and $b(s) = 1.1 + 0.4|\sin(1.256s)|$. This function $f(s, t)$ has frequently been used to model the contour of red blood cell [6].

The optimal weights τ_{N_1, N_2}^e and τ_c are computed for periodic smoothing surfaces. Here, we set the noise magnitude $\sigma = 0.3$, the smoothing parameter $\lambda = 10^{-3}$, $s_0 = t_0 = 0$ and $m_1 = m_2 = 10$ (i.e. $s_{m_1} = t_{m_2} = 10$). Fig. 1 (a) and (b) show the surface $x_{N_1, N_2}^e(s, t)$ for the case of $N_1 = N_2 = 5$ and $N_1 = N_2 = 40$ respectively, and Fig. 1 (c) shows $x_c(s, t)$.

Since these surfaces are periodic in t direction in the sense of (9), the periodicity can be more easily seen by employing a polar coordinate in terms of t . Namely, regarding s as a parameter, we introduce the following coordinates

$$(p(s, t), q(s, t)) = (x(s, t) \cos \theta(t), x(s, t) \sin \theta(t)). \quad (39)$$

Fig. 2 shows the corresponding results plotted in $o-pqs$ space. As we see from these figures, $x_{N_1, N_2}^e(s, t)$ approximates the original surface $f(s, t)$ more accurately as the number of data N_1, N_2 increases, and eventually converges to $x_c(s, t)$.

Also, Fig. 3 shows a design example of optimal closed smoothing surface and the corresponding 3-D shape model of red blood cell [7] plotted in a polar coordinate system $o-pqr$. Letting $\theta(s)$ and $\varphi(t)$ be $\theta(s) = \frac{36\pi}{180}s$ and $\varphi(t) = \frac{36\pi}{180}t$, the data \mathcal{D} is generated by sampling the following closed surface,

$$f(s, t) = \sqrt{h_1^2(\theta(s), \varphi(t)) + h_2^2(\theta(s), \varphi(t)) + h_3^2(\theta(s), \varphi(t))}$$

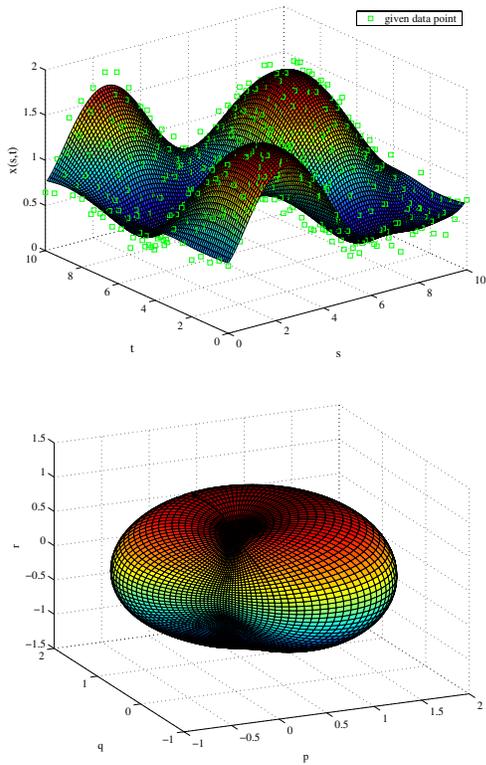


Fig. 3. Optimal closed smoothing surface (upper) and 3-D shape model of red blood cell represented in the $o-pqr$ space (lower).

with

$$\begin{aligned} h_1(\theta, \varphi) &= 0.1147 + 1.1103 \sin^2 \theta - 1.123 \sin^4 \theta \cos \theta \\ h_2(\theta, \varphi) &= 1.1087 \sin \theta \sin \varphi + 0.5 \\ h_3(\theta, \varphi) &= 1.1087 \sin \theta \cos \varphi + 0.5. \end{aligned}$$

The other parameters are set as $N_1 = N_2 = 21$, $\lambda = 10^{-3}$, $s_0 = t_0 = 0$ and $m_1 = m_2 = 10$.

In these examples, Fig. 2, in particular, indicates that the periodic spline surfaces can be used for dynamic contour modeling as we see in the following example.

B. Experimental Study

We next apply the design method of periodic splines to dynamic contour modeling of jellyfish from real digital movie file¹ with 101 [frame].

This experiment is worked out based on the idea employed in the above numerical study. In particular, we here consider the problem of modeling the jellyfish motion with deformation and translation by using a small number of image frames in the movie file.

The modeling proceeds as follows. We first assume that the i -th frame corresponds to the time $s = 0.1 \times (i - 1)$, and among 101 frames, we use only 11 frames obtained by sampling the 101 frames at every 10-th frames starting with the 1st frame. Thus, we set $N_1 = 11$ and $u_i = i - 1$, $i = 1, 2, \dots, 11$ in the data \mathcal{D} , and the interval for s as $[s_0, s_{m_1}] =$

¹Educational Image Collections, Information-technology Promotion Agency (IPA), Japan. <http://www2.edu.ipa.go.jp/gz/>

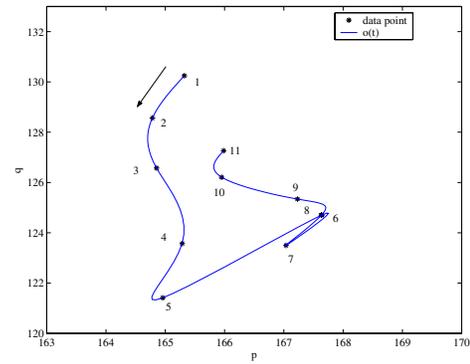


Fig. 4. Constructed translation motion of jellyfish.

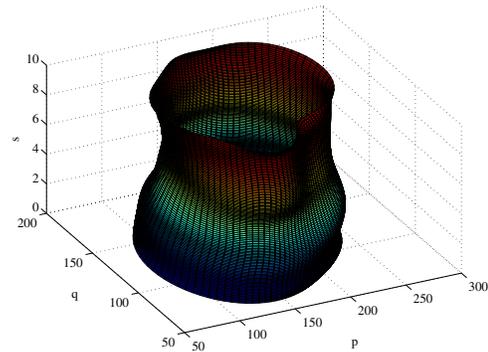


Fig. 5. Dynamic contour model of jellyfish.

$[0, 10]$ (i.e. $m_1 = 10$). Next we show how the data points v_j , $j = 1, 2, \dots, N_2$ and the data d_{ij} in \mathcal{D} are selected. The sampled i -th frame is converted to the binary image, and we compute the centroid o_i (i.e. center of mass) of target (i.e. jellyfish). Then, fixing an $o_i - p_i q_i$ plane with the origin at the centroid, and employing the so-called 'signature' [8], we computed the distance d_{ij} from the centroid to the boundary pixel at each angle $\theta = 0.2\pi v_j$ [rad] with $v_j = j - 1$, $j = 1, 2, \dots, 10$ (i.e. $N_2 = 10$). Moreover we set the interval for t as $[t_0, t_{m_2}] = [0, 10]$ (i.e. $m_2 = 10$).

We are now in the position to model the translation and deformation motions of the jellyfish. The translation motion $o(s)$ is constructed by designing smoothing curves for a set of data o_i , $i = 1, 2, \dots, 11$, and Fig. 4 shows the motion $o(s)$, $s \in [0, 10]$ in pq -plane, where the centroids o_i , $i = 1, 2, \dots, 11$ obtained from the sampled images are denoted by the corresponding numbers. On the other hand, the deformation motion is obtained by designing the periodic surface $x(s, t)$ for the set of data (u_i, v_j, d_{ij}) , $i = 1, 2, \dots, 11$, $j = 1, 2, \dots, 10$, where we set the parameters as $s_0 = t_0 = 0$, $s_{m_1} = t_{m_2} = 10$, $\lambda = 5 \times 10^{-4}$ and $w_i = \frac{1}{N_1 N_2}$. Then, by combining their results in movie frame space $o-pqs$, we get the dynamic contour model of the jellyfish as shown in Fig. 5.

Also, Fig. 6 shows some frames (26th, 46th, 66th and 86th frames) of original movie overlaid with the corresponding tomography of constructed model, i.e. the plot of $x(s, t)$

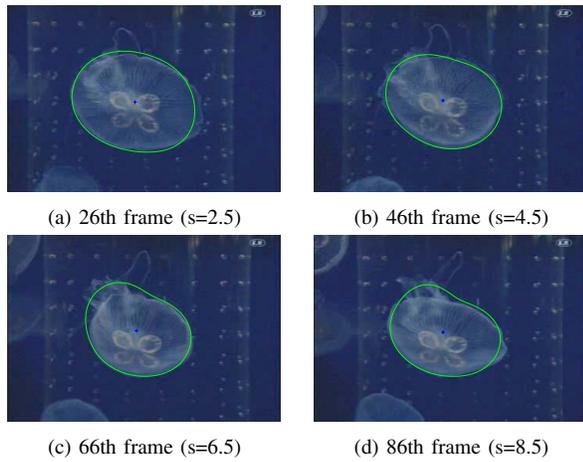


Fig. 6. Some movie frames (unused frames in \mathcal{D}) and the corresponding contour from the dynamic model.

in pq -plane for $s = 2.5, 4.5, 6.5$ and 8.5 . Although these frame data are not used for the modeling, the contour agrees with the real contour fairly precisely. Also, we confirmed by animation that the contour model for the entire motion period is in good agreement with the movie. Note that the number of sampled data used for this entire modeling is only $N_1 \times N_2 = 110$. Namely, we used 11 ($= N_1$) frames out of 101 frames and 10 ($= N_2$) data points in each frame.

In addition, such a model enables us to analyze the motion from various viewpoints. For example, the area and the smoothness from the contour model may give meaningful information for evaluating the deformation motions of jellyfish. Specifically, the area $S(s)$ and the smoothness $C(s)$ at $s \in [s_0, s_{m_1}]$ can be obtained as

$$\begin{aligned}
 S(s) &= \frac{1}{2} \int_{t_0}^{t_{m_2}} \det \begin{bmatrix} p(s,t) & q(s,t) \\ \frac{d}{dt}p(s,t) & \frac{d}{dt}q(s,t) \end{bmatrix} dt \\
 &= \frac{\pi}{t_{m_2}} \int_{t_0}^{t_{m_2}} (x(s,t))^2 dt \\
 &= \frac{\pi}{t_{m_2}} \hat{\tau}^T \left(Q_2^{(00)} \otimes B_c(s) \right) \hat{\tau}, \quad (40)
 \end{aligned}$$

$$\begin{aligned}
 C(s) &= \int_{t_0}^{t_{m_2}} \left(\frac{d^2}{dt^2} \sqrt{p^2(s,t) + q^2(s,t)} \right)^2 dt \\
 &= \int_{t_0}^{t_{m_2}} \left(\frac{d^2}{dt^2} x(s,t) \right)^2 dt \\
 &= \hat{\tau}^T \left(Q_2^{(22)} \otimes B_c(s) \right) \hat{\tau}, \quad (41)
 \end{aligned}$$

where $Q_2^{(ii)} \in \mathbf{R}^{M_2 \times M_2}$ for $i = 0, 2$ is given by (19), and $B_c(s) \in \mathbf{R}^{M_1 \times M_1}$ is defined as $B_c(s) = b_1(s)b_1^T(s)$. It is noted that the quadratic forms in (40) and (41) are easy to compute for each s since $Q_2^{(00)}$ and $Q_2^{(22)}$ are the pre-computed constant matrices and $\hat{\tau}$ is the constant vector. Fig. 7 shows the parametric representation of the computed $(C(s), S(s))$, where the points $(C(v_i), S(v_i))$, $i = 1, 2, \dots, 11$ obtained from the sampled images are denoted by the corresponding numbers. This may be helpful for evaluating the deformation motion of jellyfish.

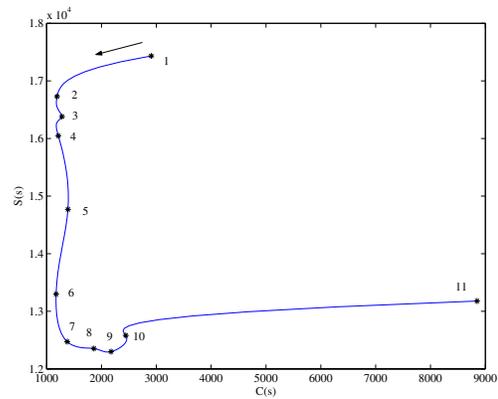


Fig. 7. Quantitative evaluation for deformation motion of jellyfish.

V. CONCLUDING REMARKS

In this paper, we developed the synthesizing scheme for modeling the contour and shape of wet material objects based on the design method optimal periodic and closed smoothing spline surfaces. The concise expressions of the optimal periodic and closed spline surfaces were derived. These enabled us to analyze the statistical and asymptotical properties of the optimal surfaces. Then, we applied the method to the problem of modeling contour and shape of red blood cell or jellyfish, and the effectiveness was verified by numerical and experimental studies.

Extending this result to higher dimensional cases, we may construct the 3D dynamic shape model of wet material objects. Such studies might be helpful to understand their whole movements involving deformation.

REFERENCES

- [1] H. Kano, H. Fujioka, M. Egerstedt and C. F. Martin, Optimal Smoothing Spline Curves and Contour Synthesis, *Proc. of the 16th IFAC World Congress*, Prague, Czech Republic, July 4-8, 2005.
- [2] A. Blake and M. Isard, *Active Contours*, Springer-Verlag, London, 1998.
- [3] A. A. Amini, Y. Chen, M. Elayyadi and P. Radeva, Tag surface Reconstruction and Tracking of Myocardial Beads from SPAMM-MRI with Parametric B-Spline Surfaces, *IEEE Trans. Medical Imaging*, vol. 20, no.2, 2001.
- [4] H. Kano, H. Nakata and C. F. Martin, Optimal curve fitting and smoothing using normalized uniform B-splines: A tool for studying complex systems, *Applied Mathematics and Computation*, 169, issue 1, pp.96-128, 2005.
- [5] H. Fujioka, H. Kano, M. Egerstedt and C. F. Martin, Smoothing Spline Curves and Surfaces for Sampled Data, *Int. J. of Innovative Computing, Information and Control*, Vol.1, No.3, pp.429-449, 2005.
- [6] A. Karlsson, J. He, J. Swartling and S. Anderson, Numerical Simulations of Light Scattering by Red Blood Cells, *IEEE Trans. Biomedical Engineering*, Vol.52, No.1, pp.13-18, 2005.
- [7] C. Pozrikidis, Numerical Simulation of the Flow-Induced Deformation of Red Blood Cells, *Annals of Biomedical Engineering*, Vol.31, pp.1194-1205, 2003.
- [8] J. P. Parker, *Algorithms for Image Processing and Computer Vision*, John Wiley & Sons, Inc., 1996.

A Method for Non-rigid 3D Deformation Fields Measurement: Application to Human Calf MR Volumetric Images

Penglin Zhang, Shinichi Hirai and Kazumi Endo

Abstract—This work is concerned with inner deformation fields measurement of non-rigid non-uniform objects from 3D Magnetic Resonance (MR) volumetric images. In order to obtain dense deformation fields of non-uniform objects, an improved robust feature matching algorithm is presented and used to obtain a matching feature point set which is the foundation of sparse local deformation fields measurement. Inner dense deformation fields can then be inferred using a linear approximate approach in an irregular tetrahedra finite element model. To test the validity of this proposed method, some experiments were designed and conducted for this paper.

I. INTRODUCTION

Since its initial use for human imaging over 20 years ago, magnetic resonance imaging (MRI) has become a widely used clinical imaging modality [1]. MRI is being increasingly employed in biomedical applications. Accordingly, MR image processing techniques have become a central issue in biomedical applications. However, most studies over the past decades have focused on MR image segmentation, registration and reconstruction, with biological tissue deformation measurement and physical parameter estimation being investigated in only a few studies.

With an ever increasing application of biomedical imaging, it is becoming more important for computer-assisted clinical diagnosis, surgery simulation and operation planning to acquire knowledge regarding the motion and deformation of biological tissue. In addition, there is much focus on the physical characteristics of tissues. In the past decade, there has been much research involving deformation measurements from MR volumetric images using elastic deformable models [2] [3] [4]. In general, deformable models can be classified into two categories: parametric and geometric active models [5]. The parametric active contours, also called snakes, were first introduced by Kass, Witkin and Terzopoulos in 1987 [6]. They were widely used in deformation estimation, segmentation, motion tracking and registration of biomedical images. Later, many researchers expanded and developed their own approaches based on this work. Lang et al. [7], Cho et al. [3] and Matuszewski et al. [2] proposed estimating the deformation of the object based on the parametric active contours. Their general idea of parametric active contours is to first define an energy function

in which the local minimum is obtained at the boundary of the object, and then to try to minimise the designed function to deform a given initial contour toward the boundary of the object to obtain deformation fields of the object.

The geometric active model was first proposed by Caselles et al. [8]. Malladi et al. [9], Caselles [10] and Chenoune et al. [5] developed different aspects of this method, but their geometric partial differential equations were proposed by Caselles et al. [8]. They used the propagation of curves and surfaces for boundary detection and motion tracking.

Although the deformable model algorithm has undergone significant development, some problems still remain. The energy model of parametric active contours is not capable of handling changes in the topology of the evolving contours when direct implementations are performed, and special, often heuristic, topology handling procedures must be used [10]. The geometric active contours do not work well for objects that have poor contrast. That is, when the object boundary is indistinct or has gaps, the contours tend to leak through the boundary [11]. In addition, it is difficult to characterise the global shape of an object with the geometric active contour algorithm.

To avoid the problems associated with the deformable model algorithm, we propose a feature tracking-based approach first to measure sparse local deformation of a non uniform non-rigid object from biomedical MR volumetric images. In our method, we automatically extract the high curvature feature points (also called points of interest) from the initial MR volumetric image, then, use the proposed matching approach to obtain their homologous positions in the final MR volumetric image. Finally, the dense deformation fields of the nonrigid non-uniform object are computed using a linear approximation approach.

This paper is organised as follows. Section 2 describes feature extraction and matching. Section 3 introduces inference of the dense deformation fields using the FF Method. Section 4 presents examples and the results of experiments. Section 5 presents a conclusion regarding the proposed approach.

II. FEATURE EXTRACTION AND MATCHING

A feature point is one of the most important factors in the present application. It is often used in various areas related to measurements based on computer vision. In this paper, the feature point matching based approach is used to obtain sparse local deformation fields of a nonrigid non-uniform object. Briefly, we first extract a certain number of feature points from the initial MR volumetric image and then match their homologous points in the final MR volumetric

Penglin Zhang is with the Faculty of Science and Engineering / School of Remote Sensing and Information Engineering, University of Ritsumeikan / University of Wuhan, 525-8577 Kusatsu, Japan zpl@se.ritsumei.ac.jp

Shinichi Hirai and Kazumi Endo are with the Faculty of Science and Engineering, Robotics, University of Ritsumeikan, 525-8577 Kusatsu, Japan hirai@se.ritsumei.ac.jp

image. The displacements corresponding to feature points are thus measured using their positions in the initial and final volumetric images, respectively. Here, high curvature points and edge points in the MR volumetric image are defined as feature points (FPS).

A. Feature Extraction

In order to automatically pre-extract a certain number of FPS, we extend the 2D Harris operator [12] to form a 3D operator. We thus obtain an auto-correlation matrix \mathbf{M} of 3D operator, which is given by

$$\mathbf{G} \otimes \mathbf{M} = \mathbf{G} \otimes \begin{pmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{pmatrix} \quad (1)$$

with eigenvector $\lambda = [\lambda_1, \lambda_2, \lambda_3]$. Where I_x , I_y and I_z are computed by convolving the image with a gradient template along the x -, y - and z -axes. The Gaussian template \mathbf{G} reduces the influence of noise. Since the eigenvector λ represents the principle curvature along x -, y - and z -axes, we define response function R_F as

$$R_F = \det(\mathbf{M}) - k * (\text{trace}(\mathbf{M}))^2 \quad (2)$$

where k is an empirical constant ($k = 0.04$ to 0.06 from Harris's suggestion), $\det(\mathbf{M})$ and $\text{trace}(\mathbf{M})$ are the determinant and trace of the matrix \mathbf{M} , respectively. Voxels whose R_F exceeds a given threshold will be regarded as FPS. From Eq.(2), we find that the selected empirical constant k affects the matching result. Thus, to avoid the influence of an improper k , in implementation, the response function also can be defined as

$$R_F = \frac{\det(\mathbf{M})}{\text{trace}(\mathbf{M})}. \quad (3)$$

B. Feature Matching

Feature matching is the foundation of our approach. The outcome may determine the accuracy of later obtained deformation fields. Our feature matching method consists of two steps: First, we perform an initial matching process using a correlation score between two cube regions around a point pair. Next, the relaxation technique is used to disambiguate matches and to obtain a potential match set (PMS). In fact, this idea is similar to the method proposed by [13] and [14]. Namely, use of a homologous algorithm.

Here, we do not describe the initial matching process as the correlation tool is a very popular algorithm used in various areas. In the course of relaxation, the cost function is defined as the average of strength of matches (SM), see [13]. The SM of a pair depends on its own correlation score $cs(\cdot)$ and the influence factor of potential matches within its neighborhood. The influence factor of potential matches includes the correlation score, residual and relative distance.

However, in later practice, we note that direction consistency of deformation fields corresponding to potential matches within the neighborhood of a given match $m(\mathbf{p}, \mathbf{p}')$ would be more important for $SM(\mathbf{p}, \mathbf{p}')$ than its residual. This is mainly because the residual is likely to be affected

by noise, whereas direction will not. Moreover, in spite of the direction of deformation fields (DODF) of non-rigid non-uniform objects being inconsistent, they are usually consistent within a local region. Therefore, if $m(\mathbf{p}_i, \mathbf{p}'_j)$ is a potential match, then its corresponding DODF must be consistent with the DODF corresponding to the potential matches within its neighborhood. In other words, if match $m(\mathbf{p}, \mathbf{p}')$ is a good match, we expect to see more potential matches within its neighborhood $\mathcal{N}(\mathbf{p}, \mathbf{p}')$, whose corresponding DODF are consistent. On the contrary, we expect to see only a few such matches, or even none at all, in their neighborhood.

Based on this idea, different to the pre-existing work, in this paper, we use DODF corresponding to feature point matches as a constraint to improve the robustness of the point matching algorithm, especially when it is used for a non-rigid non-uniform object. In this way, we yield

$$SM(\mathbf{p}_i, \mathbf{p}'_j) = cs(\mathbf{p}_i, \mathbf{p}'_j) + \alpha \sum_{k=1}^s cs(\mathbf{n}_k, \mathbf{n}'_k) \cdot w_{(\mathbf{n}_k, \mathbf{n}'_k)} \quad (4)$$

where $w_{(\mathbf{n}_k, \mathbf{n}'_k)}$ integrates the direction consistency and relative distance of potential matches $(\mathbf{n}, \mathbf{n}')$ within $\mathcal{N}(\mathbf{p}_i, \mathbf{p}'_j)$, which is given by

$$w_{(\mathbf{n}_k, \mathbf{n}'_k)} = \exp(-\mathcal{J}_k), \quad k = 1, 2, \dots, n \quad (5)$$

with notation

$$\mathcal{J}_k = \begin{cases} \text{diff}(\mathbf{p}_i, \mathbf{p}'_j; \mathbf{n}_k, \mathbf{n}'_k) & \text{if } \mathcal{O}(\mathbf{n}_k, \mathbf{n}'_k) = 0 \\ \mathcal{O}(\mathbf{n}_k, \mathbf{n}'_k) \cdot \text{diff}(\mathbf{p}_i, \mathbf{p}'_j; \mathbf{n}_k, \mathbf{n}'_k) & \text{otherwise} \end{cases}$$

where $\text{diff}(\cdot)$ represents the relative distance between $\mathbf{n}_k \Leftrightarrow \mathbf{p}_i$ and $\mathbf{n}'_k \Leftrightarrow \mathbf{p}'_j$, and $\mathcal{O}(\cdot)$ represents the constraint of the direction of potential matches. Let start position of a deformation field be the origin of Cartesian space, then, its direction can be approximated to 8 states in 2D case or 26 states in 3D case, respectively, as shown in Figure 1.

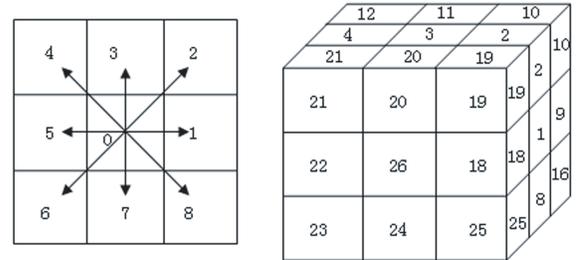


Fig. 1. The direction of deformation field (left: 2D case, right: 3D case)

Let notation $\phi_{\mathbf{p} \rightarrow \mathbf{p}'}$ be the DODF corresponding to the deformation $\mathbf{p} \rightarrow \mathbf{p}'$, then we have

$$\phi_{\mathbf{p} \rightarrow \mathbf{p}'} = \{\alpha \mid \alpha = 1, 2, 3, \dots, 26\} \quad (6)$$

where notation $\mathbf{p} \rightarrow \mathbf{p}'$ represents the deformation caused by \mathbf{p} moving to \mathbf{p}' .

Furthermore, let $A_{\mathbf{p} \rightarrow \mathbf{p}'}^{\mathbf{n} \rightarrow \mathbf{n}'}$ be the angle between $\mathbf{p} \rightarrow \mathbf{p}'$ and $\mathbf{n} \rightarrow \mathbf{n}'$, and $A_{\mathbf{p} \rightarrow \mathbf{p}'}^{\mathbf{n} \rightarrow \mathbf{n}'}$ takes 0, 1, 2 or 3 corresponding

to zero degrees, lower than 90 degrees, equal to 90 degrees and higher than 90 degrees, respectively, Then, the direction constraint factor $\mathcal{O}(\cdot)$ of a potential match in $SM(\mathbf{p}, \mathbf{p}')$ computation is defined as

$$\mathcal{O}(\mathbf{n}, \mathbf{n}') = A_{\mathbf{p} \rightarrow \mathbf{p}'}^{\mathbf{n} \rightarrow \mathbf{n}'} / 3. \quad (7)$$

The result of performing the above feature matching algorithm is that a PMS is obtained, which will be used to measure local sparse deformation fields and to infer the dense interior deformation fields of a non-uniform object.

III. INFERRING DENSE DEFORMATION FIELDS USING THE FE METHOD

In order to obtain dense interior deformation fields, an approximation approach using an FE model will be considered in this paper.

A. FE Modeling

As the first step of obtaining dense deformation fields, a finite element (FE) model of an object should be built using irregular tetrahedra. Here, since the model will be used to approximate the interior deformation fields, we use FPS from the PMS as the input in an FE model. The irregular tetrahedron algorithm used in FE modeling actually is an extension of the Delaunay triangulation irregular network (TIN) in 3D case, in brief it is given by the following pseudo code.

Begin

```

N ← the number of FPS in the initial volume;
j ← 0; // The number of formed tetrahedrons.
Empty the Triangle_List; // List of triangles which consist
of tetrahedrons.

for(i = 0 to N)
  Searching 3 FPS ⇒ first triangle; //The triangle
  //must satisfy the properties of Delaunay TIN.
for(i = 0 to N)
{
  if (The points satisfy the D_properties) Then
  {
    Form first tetrahedron;
    Triangle_List ← three new triangles;
    j ← j + 1;
  }
}
while (Triangle_List isn't empty)
{
  Pop a triangle from Triangle_List;
  for(i = 0 to N)
  {
    if(The points satisfy the D_properties) Then
    {
      Form a new tetrahedron;
      if (triangles are not in the Triangle_List);
      Triangle_List ← triangles;
      j ← j + 1;
    }
  }
}

```

End.

The notation **D_properties**, represents properties which must be satisfied in the course of modeling a tetrahedron by using a point and a triangle. These properties include: (1) The circum-sphere of a tetrahedron formed by the point and triangle must be empty. In other words, it must not contain other nodes apart from the four nodes of the formed tetrahedron. (2) The distance from a circum-sphere center to the triangle must be the minimum of all circum-spheres which may be constructed using the triangle and points.

Figure 2 illustrates the irregular tetrahedra FE model of part of a human calf.

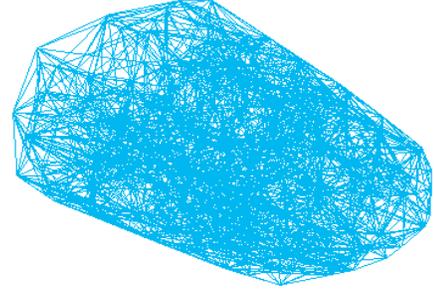


Fig. 2. Irregular tetrahedron model of part of human calf

B. Sparse local displacement measurement

Let \mathcal{D} be the displacement of a voxel before and after movement, $\mathbf{x}_1 = (x_1, y_1, z_1)$ and $\mathbf{x}_2 = (x_2, y_2, z_2)$ be the coordinate of the voxel in the initial and final MR volumetric image, respectively. Then, the displacement of a voxel is defined using the Euclidean distance between the voxel and its homologous point (Eq.8).

$$\mathcal{D} = \|\mathcal{R}(\mathbf{x}_1) - \mathbf{x}_2\| \quad (8)$$

where $\mathcal{R}(\cdot)$ represents the rotation transformation between the initial and the final MR volumetric images. In this paper, $\mathcal{R}(\cdot)$ is calculated using the unit quaternion proposed by Horn [15].

Here, we note that the translation transformation between two volumes isn't considered in the Eq.(8). This is because the translation between the initial and final MR volumetric image never affects the deformation of a nonrigid object. Thus, we don't have to consider translation but the rotation.

Through the above computation, we yield a displacement vector $\mathbf{u}_i = [dx, dy, dz]^T$ at each FEM node within the model. Displacement \mathbf{u}_i on a tetrahedron node will be used to approximate the displacement of a voxel inside the tetrahedron.

C. FE displacement

To obtain the inner dense deformation fields, we propose a piecewise linear approximation based on a 3D finite element method.

For a tetrahedron $\diamond P_i P_j P_k P_l$ within the model, let P_i be a nodal point of the tetrahedron, and $[x_i, y_i, z_i]^T$ be coordinates of point P_i . Then, the signed volume of tetrahedron $\diamond P_i P_j P_k P_l$ is given by

$$\diamond P_i P_j P_k P_l = \frac{1}{3!} \begin{vmatrix} x_i & y_i & z_i & 1 \\ x_j & y_j & z_j & 1 \\ x_k & y_k & z_k & 1 \\ x_l & y_l & z_l & 1 \end{vmatrix} \quad (9)$$

Thus, for a volume voxel P at $\mathbf{x} = (x, y, z)$ lying anywhere within an arbitrary tetrahedron $\diamond P_i P_j P_k P_l$ of the model, its displacement may be approximated by weighting the finite element's node displacements \mathbf{u}_i by their shape function

$$\mathbf{u}(\mathbf{x}) = \sum_{n=1}^4 \mathbf{u}_n N_n \quad (10)$$

where N_i is the shape function of nodal $n = (i, j, k, l)$, which is given by

$$N_n = \frac{\diamond P P_j P_k P_l}{\diamond P_i P_j P_k P_l}, \quad (11)$$

Hence, approximation displacements for all voxels \mathbf{x}_i in the volume are obtained.

IV. EXPERIMENTS AND RESULTS

Some practical examples were designed to demonstrate the capabilities of the proposed approach. All experiments were carried out using our own software developed using Visual C++, which runs on Microsoft Windows XP. And all experimental results described below were obtained on a Dell PC with a 2.80 GHz Intel Pentium D CPU and 1 GB of RAM.

In experiments, all MR volumetric images were sampled from one volunteer's calf using an MR scanner under initial and deformed cases. The initial and deformed data were sampled from the same location of the volunteer's calf. Both the initial and deformed MR volumetric images were of FOV 20×20 cm and slice gaps of 2 mm. To ensure the sufficient resolution along the z-axis, we performed linear interpolation between each two initial slices. As a result, we obtained $256 \times 256 \times 57$ voxels of initial and deformed MR volumetric images respectively. Figure 3 illustrates the volume visualization result of the data used in the experiments.

Since feature matching plays a key role in the proposed method, we compared our feature matching algorithm with the homologous approach suggested by George.Q.Chen in [14]. All the experimental results presented below are based on the PMS obtained using these two feature matching approaches.

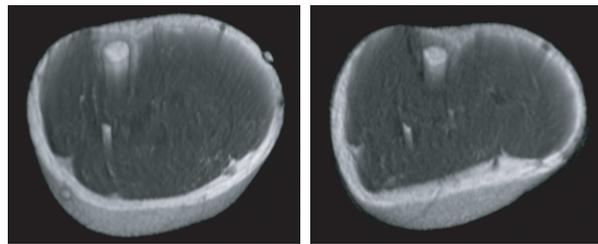


Fig. 3. Original volumes used in the experiment (left: initial volume, right: deformed volume)

Feature matching is the first phase of deformation fields measurement. In this phase, the size of a match cube is set as $9 \times 9 \times 3$ pixels. Whereas, the size of a search cube is dynamic varies depending on the distance of the feature point far from the moment center of the slice which the feature point lies on. The initial size of a search cube is set as $17 \times 17 \times 3$ voxels. Table I show the results obtained using different feature matching methods.

After obtaining PMS, a finite element model of a nonrigid object is reconstructed accordingly, and the displacements of all nodes in the model are also computed by using approaches introduced in section III. Figure 4 shows the displacement of each FEM node.

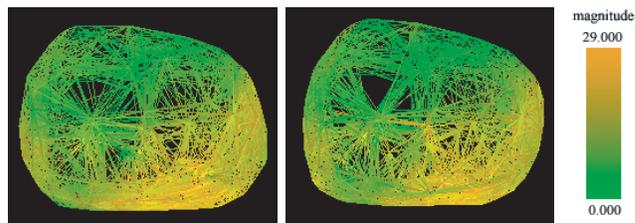


Fig. 4. Displacement of an FEM node (Left: PMS obtained using the proposed method, there are 4344 tetrahedra in total; Right: PMS obtained using the method proposed in [14], there are 4798 tetrahedra in total)

Figure 5 illustrates the visualization result of dense deformation fields. Here, the dense deformation vector fields are first computed using the above linear approximation approach by our own software, then, it is saved into a file as Metalmage format and visualized in free software Paraview. To facilitate observation of the inner deformation of different regions, only 30000 vectors are shown in the figure.

To evaluate the accuracy of the deformation fields, the evaluation approach suggested in [16] is used in our experiment. That approach uses the root mean squared (RMS) of residual differences (Eq. 12) to evaluate the quality of the

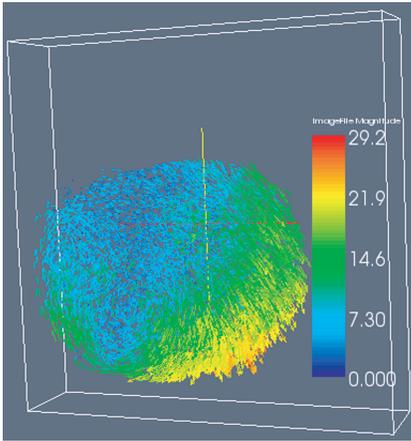


Fig. 5. Dense deformation fields (Deformation magnitudes are mapped using rainbow color coding, with dark blue corresponding to 0-pixel, and dark red corresponding to 30-pixels deformation magnitude.)

registration result using deformation fields.

$$\mathbf{E}_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{\mathbf{x} \in \Omega} (I_a(\mathbf{x}) - I_r(\mathbf{x}))^2} \quad (12)$$

In implementation, we deform the initial volume to obtain a computation deformation volume using measured dense deformation fields. Thereafter, an RMS algorithm is applied between the actual deformed volume and two computation results. Table I gives the comparison results. It is noted that, due to rotation and translation transformation along the z -axis, after resampling, some slices are out of the boundary of z . Thus, in this experiment, slices from 10 to 50 are used to compute \mathbf{E}_{RMS} .

Finally, to observe if the deformation fields are valid or not, Figure 6 presents images of the calf part actual deformation slices, computation slices and the differences between them. The computation slices were re-sampled results of the initial volume dependent on the measured dense deformation fields.

V. CONCLUSIONS AND FUTURE WORK

A. Conclusions

This paper proposed a feature match based approach to measure the deformation fields of a non-uniform nonrigid object from 3D MR volumetric images. Main contributions include:

(1) Presentation of a new approach in feature matching based deformation fields measurement from MR volumetric images. Different from many registration approaches currently used to measure deformation fields, this paper

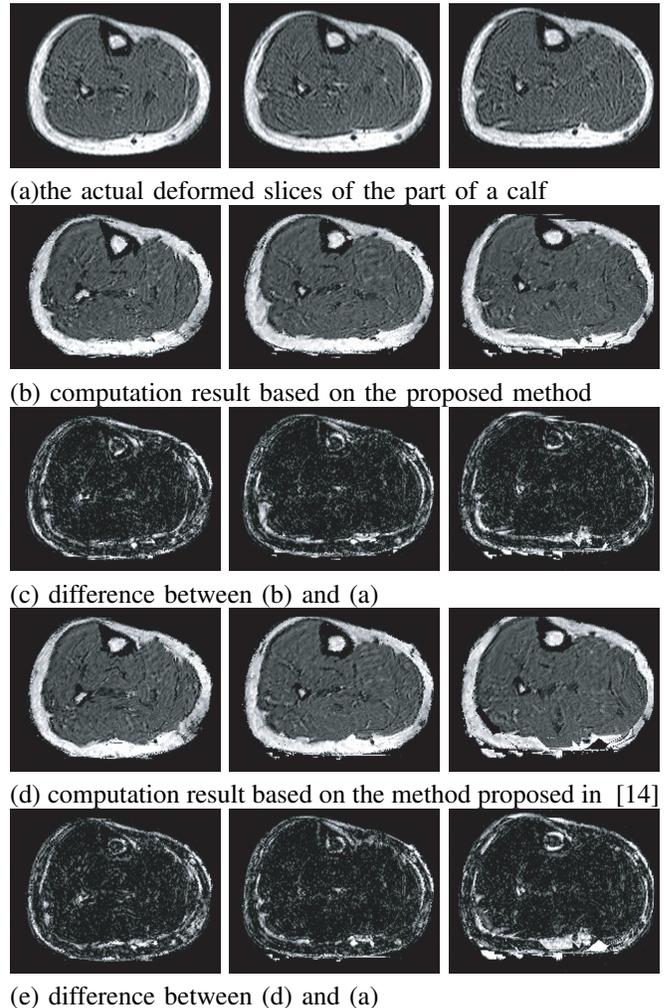


Fig. 6. Resample and Difference (From left to right, slice numbers are 15, 30 and 45, respectively)

proposed inferring dense deformation fields dependent on the sparse local deformation fields corresponding to the feature points in the MR volumetric image. The advantages include: (a) The computation cost of finite feature points is lower than that of voxel-by-voxel as used by most registration approaches. (b) The proposed approach is noise insensitive compared with traditional registration approaches.

(2) Improvement to the robustness of a feature matching algorithm for a nonrigid non-uniform object, local direction consistency and relative distance are introduced into the feature matching algorithm as constraints, making the algorithm more suitable for application to nonrigid objects than originally proposed in [13] and [14].

Our preliminary experimental results shown in section IV

TABLE I
COMPARISON RESULT FOR DIFFERENT FEATURE MATCHING APPROACHES

Approaches	Point Numbers in IV	Point Numbers in FV	Potential Matches	Tetrahedra	<i>RMS</i>
Approach in this paper	1000	5000	771	4344	26.004284253441
Approach in [14]	1000	5000	827	4798	26.351873220353

(IV: Initial volume; FV: Deformed volume.)

reveal that the improved feature matching method is efficient. Moreover, the differences between the actual slices and the computation slices (Figure 6) indicate that we can deform the initial volume to the deformation volume using obtained deformation fields. It further reveals that the deformation fields are valid.

Further improvement includes applying the proposed approach to other data-sets in order to further test the robustness of this method.

B. Future Works

Physical parameters estimation is our final goal. Thus our next work should be to estimate the physical parameters of non-rigid non-uniform objects based on the Finite Element Model (FEM) shown in Eq. (13).

$$\mathbf{f} = \mathbf{k}\mathbf{d} \quad (13)$$

where vector \mathbf{f} represents the force acting on the FEM nodes, \mathbf{k} represents the stiffness matrix which depends on the physical parameters of the object, and \mathbf{d} is the displacement vector corresponding to the FEM nodes.

It is obvious that the deformation fields obtained using the approach presented in this paper will act as the input for physical parameters estimation.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of rheology research group.

REFERENCES

- [1] S. J. Riederer, "Current technical development of magnetic resonance imaging," *ENGINEERING IN MEDICINE AND BIOLOGY. IEEE.*, vol. 19, no. 5, pp. 34–41, 2000.
- [2] B. J. Matuszewski, J. K. Shen, and L. K. Shark, "Estimation of internal body deformations using an elastic registration technique," *Proceedings of the International Conference on Medical Information Visualisation-BioMedical Visualisation*, pp. 15–20, 2006.
- [3] J. Cho and P. J. Benkeser, "Elastically deformable model-based motion-tracking of left ventricle," *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, vol. 1, no. 3, pp. 1925–1928, 2004.
- [4] X. Papademetris, A. J. Sinusas, D. P. Dione, R. T. Constable, and J. S. Duncan, "Estimation of 3-d left ventricular deformation from medical images using biomechanical models," *Transactions on Medical Imaging. IEEE*, vol. 21, no. 7, pp. 786–800, July 2002.
- [5] Y. Chenoune, E. Delechelle, E. Petit, T. Goissen, J. Garot, and A. Rahmouni, "Segmentation of cardiac cine-mr images and myocardial deformation assessment using level set methods," *Computerized Medical Imaging and Graphics*, vol. 29, no. 8, pp. 607–616, Dec. 2005.
- [6] M. Kass and A. Witkin, "Snakes: active contours models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, Jan. 1987.
- [7] J. Lang, D. K. Pai, and R. J. Woodham, "Robotic acquisition of deformable models," in *Proc. ICRA '02*, vol. 1, 2002, pp. 933 – 938.
- [8] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische Mathematik*, vol. 66, no. 1, pp. 1–31, Dec. 1993.
- [9] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: a level set approach," *Transactions on Pattern Analysis and Machine Intelligence. IEEE*, vol. 17, no. 2, pp. 158–175, Feb. 1995.
- [10] V. Caselles, "Geometric models for active contours," *Image Processing. Proceedings. International Conference on*, vol. 3, pp. 9–12, Oct. 1995.
- [11] F. Huang and J. Su, "Face contour detection using geometric active contours," *Proceedings of the 4th World Congress on Intelligent Control and Automation*, June 2002.
- [12] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of The Fourth Alvey Vision Conference*, pp. 147–151, 1988.
- [13] P. L. Zhang, S. Hirai, K. Endo, and S. Morikawa, "Local deformation measurement of biological tissues based on feature tracking of 3d mr volumetric images," *2007 IEEE/ICME International Conference on Complex Medical Engineering*, May 2007.
- [14] G. Q. Chen, "Robust point feature matching in projective space," *Computer Vision and Pattern Recognition, 2001. CVPR 2001*, vol. 1, pp. 717–722, 2001.
- [15] B. K. Horn, "Closed form solution of absolute orientation using unit quaternions," *Optical Society of America A: Optics, Image Science, and Vision*, vol. 4, no. 4, pp. 629–642, 1987.
- [16] P. Rogelj, S. Kovacic, and J. C. Gee, "Validation of a non-rigid registration algorithm for multi-modal data," *Medical Imaging 2002: Image Processing*, pp. 299–307, 2002.

Modeling of Deformable Linear Objects

Presentation 1: *Modeling of Linear and Belt Object Deformation Based on Differential Geometry*

Pages: 15–24

Presentation 2: *Rope Knotting and Unknotting with Haptic Feedback*

Pages: 25–32

Modeling of Linear and Belt Object Deformation Based on Differential Geometry

Hidefumi Wakamatsu, Eiji Arai, and Shinichi Hirai

Abstract— A modeling of linear and belt object deformation based on differential geometry is proposed. It is difficult to manipulate linear objects such as cables and belt objects such as film circuit boards successfully by a robot because they can be easily deformed and must be appropriately deformed for assembly. Therefore, estimation of linear/belt object deformation is required. First, a modeling method to describe linear object deformation including flexure, torsion, and extension is proposed by extending differential geometry. Next, some computational results are compared with experimental results to demonstrate the accuracy of our proposed model. Knotted/knitted shapes of a linear object are also computed to show the versatility of this method. Finally, this method is applied to belt object deformation and its validity is verified with measuring experiments.

I. INTRODUCTION

Many manipulative operations deal with deformable linear objects such as wires, cords, and threads with flexural, torsional, and extensional deformations in 3D space. For example, wires and cables are manipulated in the manufacturing of electrical apparatuses and automobiles. In addition, flexible thin objects with belt-like shape such as film circuit boards and flat cables are installed into various electronic devices. Linear/thin objects are defined as objects that are much larger/smaller along one of the orthogonal directions than along the other two. Modeling of linear/thin object deformation is required for many purposes, including planning of manipulative operations and design of products.

There has been a great deal of research regarding the modeling of linear/thin object deformation. Linear objects have been approximated using beams in the engineering community; models exist to describe small deflection of beams[1], and also large deformation using nonlinear beam finite elements[2]. With respect to thin objects, Kirchhoff theory for thin plates and Reissner-Mindlin theory for thick plates have been used[3]. For very thin plates, the inextensional theory was proposed[4]. In this theory, it is assumed that the middle surface of a plate is inextensional, that is, the surface of the plate is developable. Displacement of linear/thin objects can be calculated using FEM based on these models/theories. However, the high aspect ratio of linear objects such as wire and thread, and thin objects such as paper and cloth often causes instability in computation of deformed shapes. Thus, various modeling techniques have

been adapted for linear or thin objects. For example, the deformed shape of a thread suspended by two points has been analyzed using calculus of variations, and shown to be described by a catenary[5]. The deformation of clothes has also been described using catenaries[6]. Nonlinear shell theory has been applied to the modeling of fabric deformation[7]. A particle-based model of cloth has been proposed for drape simulation[8]. Implicit numerical integration has been introduced to the particle-based cloth model to reduce computation time[9]. Fast algorithms have been introduced to describe linear object deformation using the Cosserat formulation[10]. Cosserat elements possess six degrees of freedom; three for translational displacement and three for rotational displacement. Flexure, torsion, and extension of a linear object can be described by use of Cosserat elements. In differential geometry, curved lines in 2D or 3D space have been studied to describe their shapes mathematically[11]. Moll et al. have proposed a method to compute the stable shape of a linear object under some geometrical constraints quickly based on differential geometry[12].

In this paper, a modeling of linear object deformation based on differential geometry is described and applied to a structure composed of linear objects such as a fabric and a rectangular thin object, namely, a belt object. First, differential geometry is extended to describe linear object deformation including flexure, torsion, and extension. Next, some computational results are shown to demonstrate the feasibility of the proposed modeling technique and they are compared with experimental results to demonstrate the accuracy of the model. Moreover, knotted/knitted shapes of a linear object are computed using our proposed approach. Finally, this approach is applied to deformation of an inextensible rectangular belt object and the validity of the object model referred to as *fishbone model* is verified with measuring experiments.

II. MODELING OF LINEAR OBJECT DEFORMATION

A. Differential geometry coordinates

In this section, the deformation of a linear object in 3D space will be formulated. As described by Frenet-Serret formulas in differential geometry, any curve in 3D space can be specified by Frenet frame field[11]. But, extension along a linear object and torsion around its central axis cannot be described in the Frenet frame field. Instead, we will specify the relationship between two frame fields defined in natural and deformed states of a linear object.

H. Wakamatsu and E. Arai are with Dept. of Materials & Manufacturing Science, Graduate School of Eng., Osaka Univ., 2-1 Yamadaoka, Suita, Osaka 565-0871, Japan {wakamatu, arai}@mapse.eng.osaka-u.ac.jp

S. Hirai is with Dept. of Robotics, Ritsumeikan Univ., 1-1-1 Noji Higashi, Kusatsu, Shiga 525-8577, Japan hirai@se.ritsumei.ac.jp

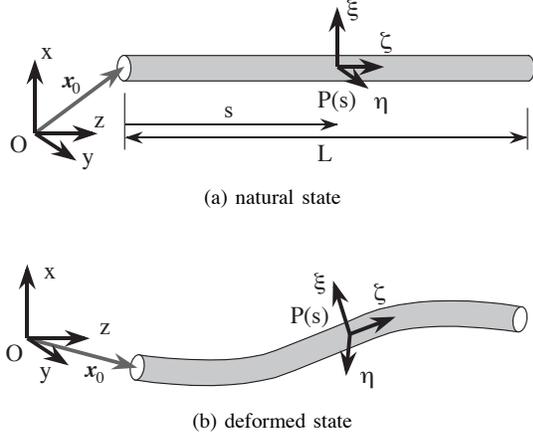


Fig. 1. Coordinate systems describing linear object deformation

First, we assume that the shape of cross-section of a linear object is not changed, namely, deformation in any direction perpendicular to its central axis is negligible. Let L be the length of the object and s be the distance from one end point of the object along its central axis. Let $P(s)$ be the point on the object at distance s . In order to describe the deformation of a linear object, the global space coordinate system and the local object coordinate systems at individual points on the object are introduced as shown in Fig.1. Let $O - xyz$ be the coordinate system fixed in space and $P - \xi\eta\zeta$ be the coordinate system fixed at an arbitrary point $P(s)$ on the object. Assume that the object is straight along the z -axis in its natural state whereby the object has no deformation. Select the direction of coordinates so that the ξ -, η -, and ζ -axes are parallel to the x -, y -, and z -axes, respectively, in the natural state. Deformation of the object is then represented by the relationship between the local coordinate system $P - \xi\eta\zeta$ at each point on the object and the global coordinate system $O - xyz$. This is referred to as *differential geometry coordinate representation*.

Let us describe the orientation of the local coordinate system with respect to the space coordinate system by use of Eulerian angles, $\phi(s)$, $\theta(s)$, and $\psi(s)$. The rotational transformation from the coordinate system $P - \xi\eta\zeta$ to the coordinate system $O - xyz$ will be expressed by the following rotation matrix:

$$A(\phi, \theta, \psi) = \begin{bmatrix} C_\theta C_\phi C_\psi - S_\phi S_\psi & -C_\theta C_\phi S_\psi - S_\phi C_\psi & S_\theta C_\phi \\ C_\theta S_\phi C_\psi + C_\phi S_\psi & -C_\theta S_\phi S_\psi + C_\phi C_\psi & S_\theta S_\phi \\ -S_\theta C_\psi & S_\theta S_\psi & C_\theta \end{bmatrix}. \quad (1)$$

For the sake of simplicity, for example, $\cos \theta$ and $\sin \theta$ are abbreviated as C_θ and S_θ , respectively. Note that the Eulerian angles depend on distance s . Let ξ , η , and ζ be unit vectors along the ξ -, η -, and ζ -axes, respectively, at point $P(s)$. These unit vectors are given by the first, second, and third columns of the rotation matrix, respectively. Namely,

$$A(\phi, \theta, \psi) = [\xi \mid \eta \mid \zeta].$$

Let us describe the curvature of a deformable linear object

and its torsional angle in order to express flexural and torsional deformations of the object. Let ω_ξ , ω_η , and ω_ζ be infinitesimal ratios of rotational angles around the ξ -, η -, and ζ -axes, respectively, at point $P(s)$ to distance s . They correspond to differentiation of rotational angles around these three axes with respect to distance s and they are described as follows:

$$\begin{bmatrix} \omega_\xi \\ \omega_\eta \\ \omega_\zeta \end{bmatrix} = \begin{bmatrix} -S_\theta C_\psi \\ S_\theta S_\psi \\ C_\theta \end{bmatrix} \frac{d\phi}{ds} + \begin{bmatrix} S_\psi \\ C_\psi \\ 0 \end{bmatrix} \frac{d\theta}{ds} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \frac{d\psi}{ds}. \quad (2)$$

Let κ and ω be the curvature and the torsional angle at point $P(s)$, respectively. The curvature and torsional angle can then be described using infinitesimal angle ratios as follows:

$$\kappa^2 = \omega_\xi^2 + \omega_\eta^2 = \left(\frac{d\theta}{ds} \right)^2 + \left(\frac{d\phi}{ds} \right)^2 \sin^2 \theta, \quad (3)$$

$$\omega^2 = \omega_\zeta^2 = \left(\frac{d\phi}{ds} \cos \theta + \frac{d\psi}{ds} \right)^2. \quad (4)$$

Note that the curvature κ and the torsional angle ω both depend on distance s .

In order to express the extensional deformation of a linear object, a strain at each point $P(s)$ is introduced. Let ε be extensional strain at point $P(s)$ on a linear object along its central axis. It turns out that the unit vector along the ζ -axis at the natural state can be transformed into the following vector due to the object deformation:

$$(1 + \varepsilon) \zeta(s) = (1 + \varepsilon) \begin{bmatrix} S_\theta C_\phi \\ S_\theta S_\phi \\ C_\theta \end{bmatrix}. \quad (5)$$

Let $\mathbf{x}(s) = [x(s), y(s), z(s)]^T$ be the position vector of point $P(s)$. The position vector can be computed by integrating vector $(1 + \varepsilon) \zeta(s)$. Namely,

$$\mathbf{x}(s) = \mathbf{x}_0 + \int_0^s (1 + \varepsilon) \zeta(s) ds, \quad (6)$$

where $\mathbf{x}_0 = [x_0, y_0, z_0]^T$ is the position vector at the end point $P(0)$.

From the above discussion, it is found that the geometrical shape of a deformed linear object can be represented by four functions, namely, Eulerian angles ϕ , θ , and ψ , and extensional strain ε . Note that each function depends upon parameter s .

B. Potential energy and geometric constraints

In this paper, the variational principle in statics will be applied to the modeling of linear object deformation. According to this principle, under the imposed constraints, the potential energy of a linear object attains its minimum value in its stable deformed state.

First, the potential energy of a linear object is formulated. Applying Bernoulli and Navier's assumption, it turns out that the potential energy U can be described as follows:

$$U = U_{\text{flex}} + U_{\text{tor}} + U_{\text{ext}} + U_{\text{grav}} \quad (7)$$

where U_{flex} , U_{tor} , and U_{ext} represent the flexural, torsional, and extensional energy of the object, respectively, and U_{grav} denotes its gravitational energy.

The object's total flexural energy U_{flex} and total torsional energy U_{tor} can be computed by integrating, respectively, flexural energy and torsional energy at point $P(s)$ over the object. Assuming that the flexural energy and the torsional energy are proportional to the bending moment and twisting moment at each point $P(s)$, respectively, the energies can be described as follows:

$$U_{\text{flex}} = \frac{1}{2} \int_0^L R_f \kappa^2 ds, \quad (8)$$

$$U_{\text{tor}} = \frac{1}{2} \int_0^L R_t \omega^2 ds \quad (9)$$

where R_f and R_t represent the flexural and torsional rigidity at point $P(s)$, respectively. Note that R_f and R_t may vary with respect to distance s . Assuming that the extensional energy is proportional to the extensional strain at each point $P(s)$, extensional energy U_{ext} is given by

$$U_{\text{ext}} = \frac{1}{2} \int_0^L R_e \varepsilon^2 ds \quad (10)$$

where R_e denotes the extensional rigidity of the object, which may depend on distance s . Assuming that gravity forces act along the x -axis, the gravitational energy is given by

$$U_{\text{grav}} = \int_0^L D x ds \quad (11)$$

where D represents weight per unit length of the object. The quantity D may also vary with distance s . Thus, the potential energy can also be represented in terms of the four variables $\phi(s)$, $\theta(s)$, $\psi(s)$, and $\varepsilon(s)$.

The interaction between the linear object and other objects such as fingertips or obstacles imposes geometric constraints on the linear object. Let us formulate these geometric constraints. The relative position between two points on the object is often controlled during a manipulative operation of the object. Consider a constraint that specifies the positional relationship between two points on the object. Let $\mathbf{l} = [l_x, l_y, l_z]^T$ be a predetermined vector describing the relative position between two operational points, $P(s_a)$ and $P(s_b)$. Recall that the spatial coordinates corresponding to distance s are given by (6). Thus, the following equation must be satisfied:

$$\mathbf{x}(s_b) - \mathbf{x}(s_a) = \mathbf{l}. \quad (12)$$

The orientation at one point on the object is often controlled during an operation as well. This constraint is simply described as follows:

$$A(\phi(s_c), \theta(s_c), \psi(s_c)) = A(\phi_c, \theta_c, \psi_c), \quad (13)$$

where ϕ_c , θ_c , and ψ_c are predefined Eulerian angles at one operational point $P(s_c)$.

Contact between a linear object and rigid obstacles in operation space also yields other geometric constraints. Note

that any point on the object must be located on or outside each obstacle. Let us describe the surface of an obstacle fixed in space by function $f(\mathbf{x}) = 0$. Assume that the value of the function is positive inside the obstacle and negative outside it. The condition that a linear object is not interfered by this obstacle is then described as follows:

$$f(\mathbf{x}(s)) \leq 0, \quad \forall s \in [0, L], \quad (14)$$

where $\mathbf{x}(s)$ is described in (6). Note that the condition that an object is not interfered by obstacles is described by a set of inequalities since mechanical contacts between the objects constrain the motion of the object unidirectionally.

Furthermore, self-interaction of a linear object should be considered. Assume that the cross section of a linear object is circular. Let $r(s)$ be the radius of the cross section at point $P(s)$. Then, in order to avoid interference with itself, a linear object must satisfy the following condition:

$$|\mathbf{x}(s_i) - \mathbf{x}(s_j)| \geq r(s_i) + r(s_j), \\ \forall s_i, s_j \in [0, L], \text{ s.t. } |s_i - s_j| \geq r(s_i) + r(s_j). \quad (15)$$

From the above discussion, it is found that the geometric constraints imposed on a linear object are given by not only equational constraints such as (12) and (13) but also inequality constraints such as (14) and (15). The deformed shape of the object is, therefore, determined by minimizing the potential energy described in (7) under these geometric constraints imposed on the object. Namely, computation of the deformed shape of an object results in a variational problem under equational and inequality constraints.

III. VERIFICATION OF LINEAR OBJECT MODEL

A. Computation algorithm

Computation of the deformed shape of a linear object results in a variational problem as mentioned in the previous section. One method to solve a variational problem is Euler's approach, which is based on the stationary condition in function space. Recall that the geometric constraints resulting from mechanical contacts are unidirectional and mathematically describable by inequalities such as (14) and (15). These conditions are nonholonomic constraints[13]. Thus, the shape of an object that minimizes internal energy does not necessarily satisfy the stationary condition. This implies that Euler's approach, which is based on the stationary condition, is not applicable.

In this paper, an algorithm based on Ritz's method[14] and a nonlinear programming technique will be developed. Let us express functions $\phi(s)$, $\theta(s)$, $\psi(s)$, and $\varepsilon(s)$ by linear combinations of basic functions $e_1(s)$ through $e_n(s)$:

$$\phi(s) = \sum_{i=1}^n a_i^\phi e_i(s) \triangleq \mathbf{a}^\phi \cdot \mathbf{e}(s), \quad (16)$$

$$\theta(s) = \sum_{i=1}^n a_i^\theta e_i(s) \triangleq \mathbf{a}^\theta \cdot \mathbf{e}(s), \quad (17)$$

$$\psi(s) = \sum_{i=1}^n a_i^\psi e_i(s) \triangleq \mathbf{a}^\psi \cdot \mathbf{e}(s), \quad (18)$$

$$\varepsilon(s) = \sum_{i=1}^n a_i^\varepsilon e_i(s) \triangleq \mathbf{a}^\varepsilon \cdot \mathbf{e}(s), \quad (19)$$

where \mathbf{a}^ϕ , \mathbf{a}^θ , \mathbf{a}^ψ , and \mathbf{a}^ε are vectors consisting of coefficients corresponding to functions $\phi(s)$, $\theta(s)$, $\psi(s)$, and $\varepsilon(s)$, respectively, and vector $\mathbf{e}(s)$ is composed of basic functions $e_1(s)$ through $e_n(s)$. Substituting the above equations into (7), potential energy U is described by a function of coefficient vectors \mathbf{a}^ϕ , \mathbf{a}^θ , \mathbf{a}^ψ , and \mathbf{a}^ε . The geometric constraints are also described by conditions involving the coefficient vectors. In addition, discretizing (14) and (15) by dividing interval $[0, L]$ into N small intervals yields a finite number of conditions. As a result, a set of the geometric constraints is expressed by equations and inequalities in terms of the coefficient vectors.

Consequently, the deformed shape of a linear object can be derived by computing a set of coefficient vectors \mathbf{a}^ϕ , \mathbf{a}^θ , \mathbf{a}^ψ , and \mathbf{a}^ε that minimizes the potential energy under the geometric constraints. This minimization problem under equality and inequality constraints can be solved by the use of a nonlinear programming technique such as the multiplier method[15]. In this method, a minimization problem under geometric constraints is converted into an unconditional minimization problem with Lagrange multipliers. The Lagrange multipliers denote the components of reaction forces corresponding to individual geometric constraints. The shape of the deformed object corresponding to a set of coefficient vectors can be computed by (6).

B. Computation of Linear Object Deformation

In this section, numerical examples will demonstrate how the proposed method computes the deformed shape of a linear object. The following set of basic functions are used in the computation:

$$\begin{aligned} e_1 &= 1, & e_2 &= s, \\ e_{2i+1} &= \sin \frac{2\pi i s}{L}, \\ e_{2i+2} &= \cos \frac{2\pi i s}{L}, \quad (i = 1, 2, 3, 4). \end{aligned} \quad (20)$$

We apply the multiplier method and BFGS formula in the quasi-Newton method to the nonlinear optimization. The multiplier method converts a minimization problem with geometric constraints into an unconditional minimization problem. The BFGS formula solves the converted unconditional minimization problem. All optimizations start from the natural state of a linear object.

Let us consider deformation of a linear object with flexure and torsion. The potential energy of the object is then given by the sum of the flexural and torsional energies of the object; $U = U_{\text{flex}} + U_{\text{tor}}$. Let us align the central axis at both end points of a linear object in the initial state. Then, let us move one end point along this axis in order to shorten the distance between the two end points specified by l . Computed shapes of a linear object of length L are shown in Fig.2. Values of the distance l corresponding to the computed shapes are $0.6L$, $0.5L$, $0.4L$, $0.3L$, $0.2L$, and $0.1L$. The shape of a

linear object changes from a knot-free shape into a one-knot shape while the distance between the two end points decreases, as shown in the figure. In a one-knot shape, the object has not only flexural deformation but also torsional deformation. This topological shape transition occurs when the potential energy of a one-knot shape becomes smaller than that of a knot-free shape. Our proposed approach can simulate such shape transition.

C. Measurement of Linear Object Deformation

In this section, the computation results will be experimentally verified by measuring the deformed shape of a linear object. We have measured the shape of a metal wire 871mm long. The wire's flexural rigidity R_f and the torsional rigidity R_t are $6.6 \times 10^{-4} \text{ N}\cdot\text{m}^2$ and $2.3 \times 10^{-4} \text{ N}\cdot\text{m}^2$, respectively. Weight D per unit length of the wire is $1.0 \times 10^{-2} \text{ N/m}$. Two manipulators control the position and orientation of both end points of the wire. In the initial state, one end point is rotated by ω_0 while keeping the wire straight. Next, the distance between the two end points l_z is decreased by controlling the motion of the two manipulators. Then, the object is both flexed and twisted. This implies that its shape is no longer limited to one plane. The shape of the deformed wire is measured by two cameras. The optical axes of the cameras intersect at one point at right angles. Let one optical axis be the x -axis and the other axis be the y -axis. The projections of the deformed shape onto the z - x and z - y planes can then be directly measured by the two cameras.

Fig.3 shows the computed and measured shapes of a deformed wire at $\omega_0 = \pi \text{ rad}$. The solid and dotted lines represent the computed and measured values, respectively. From the measured values, the distances between the two end points along the x -axis and the y -axis; l_x and l_y , and the orientations at the end points; $A(\phi(0), \theta(0), \psi(0))$ and $A(\phi(L), \theta(L), \psi(L))$ have been estimated. The estimated values have been used in the computation of the deformed shapes of a wire. The gravitational effect is assumed to be negligible. The difference between the computed and experimental values along the x - and y -axes is 50mm at most. The ratio of the difference in the wire length is approximately 6%. This difference may be caused by accuracy of estimation of the orientations at the end points. We can easily measure the direction of vector ζ , namely, the direction of the central



Fig. 2. Computational result of topological shape transition

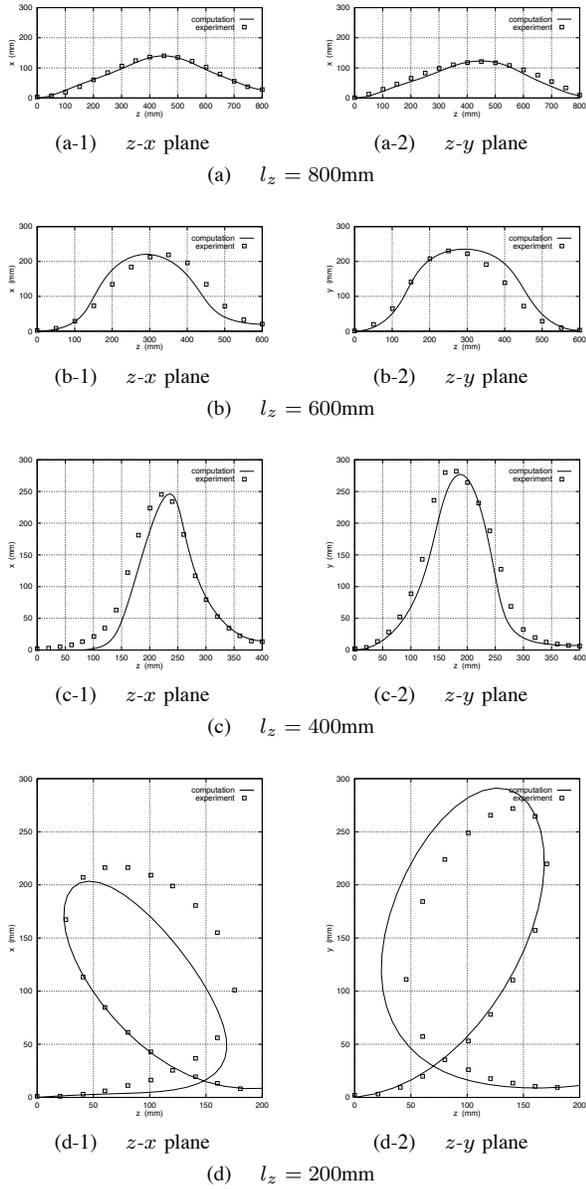


Fig. 3. Computed and measured 3D shapes of wire

axis, but it is difficult to estimate directions of vectors ξ and η .

IV. APPLICATION TO LINEAR OBJECT STRUCTURE

A. Knotted shape

Deformable linear objects such as threads and yarns are used to suture or ligate organs and tissues in surgery and to weave or knit textiles. For training of surgery or textile design, modeling of such objects is required. Knot tying of a thread has been simulated using a particle-based model of the thread[16]. Shi et al. has been developed a real-time simulator with visual and force feedback for suturing, that is, knotting and unknotting threads[17]. The deformed shape of threads in a fabric has been described geometrically[18]. Wada et al. modeled plain knitted fabrics as combination of

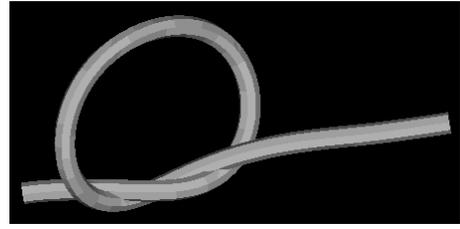


Fig. 4. Computational result of overhand knot

yarns deformed in 2D space[19]. In this section, we will apply our method of linear object deformation to more complex shapes such as knots and knitted fabrics to demonstrate its versatility.

First, let us consider the knotted shape of a linear object. A knotted linear object has some crossings. For example, an overhand knot has three crossings. The object contacts with itself at each crossing. Therefore, self-contact of a linear object must be considered to express the knotted shapes. Assume that the cross section of a linear object is circular. Let $r(s)$ be the radius of the cross section at point $P(s)$. Let s_i^u and s_i^l be distance at the upper and the lower crossing point at i -th crossing, respectively. To contact with itself at the crossing, the following equations must be satisfied:

$$z(s_i^u) - z(s_i^l) = 0, \quad (21)$$

$$x(s_i^u) - x(s_i^l) = 0, \quad (22)$$

$$y(s_i^u) - y(s_i^l) = r(s_i^u) + r(s_i^l). \quad (23)$$

The sequence of crossing points along the object must be maintained. If the upper crossing point is closer to the left endpoint $P(0)$ than the lower crossing point, the following condition must be satisfied:

$$s_i^u < s_i^l. \quad (24)$$

Note that s_i^u and s_i^l are parameters which determine the stable knotted shape as well as coefficients α described by (16) through (19). Then, the knotted shape can be derived by computing α , s_i^u , and s_i^l which minimize the potential energy under constraints including (22) through (24). Fig.4 shows the computational result of an overhand knot.

B. Knitted shape

A fabric is a structure composed of linear objects such as yarns. Fig.5 illustrates a plain knitted fabric. Assume that the radius of all yarn is constantly equal to r . As shown in this figure, the same structure called a *knitted loop* is arranged in the plain knitted fabric. The column and the row directions are referred to as *wale* and *course*, respectively. Let l_w and l_c be loop intervals in wale direction and in course direction, respectively. One loop, which is hatched, has eight crossings 1 through 8 as shown in Fig.5. Let s_1 through s_8 be distance at corresponding crossing points, respectively. Recall that a plain knitted fabric has iterative structure. So, we assume that the shape of the fabric can be represented by repetitions of the same shape of one loop. Then, the following conditions

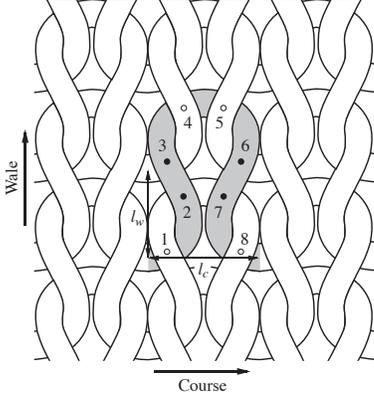


Fig. 5. Loop structure of plain knitted fabric

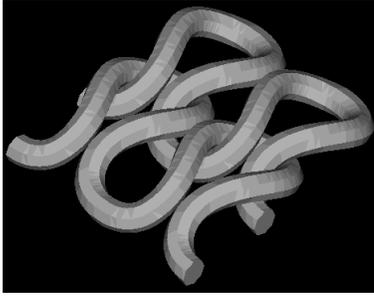


Fig. 6. Computational result of plain knitted fabric

must be satisfied:

$$z(s_i) - z(s_{i+2}) = 0, \quad (i = 1, 2, 5, 6), \quad (25)$$

$$x(s_i) - x(s_{i-2}) = l_w, \quad (i = 3, 4), \quad (26)$$

$$x(s_i) - x(s_{i+2}) = l_w, \quad (i = 5, 6), \quad (27)$$

$$y(s_i) - y(s_{i-2}) = 2r, \quad (i = 3, 7), \quad (28)$$

$$y(s_i) - y(s_{i+2}) = 2r, \quad (i = 2, 6), \quad (29)$$

$$s_i < s_{i+1}, \quad (i = 1, \dots, 7). \quad (30)$$

Note that s_1 through s_8 are determined by minimization of the potential energy of one loop. Fig.6 shows the computational result of a plain knitted fabric. In this figure, one loop is arranged in 2×2 matrix. Thus, various deformation of linear objects can be computed using our proposed method.

V. MODELING OF BELT OBJECT DEFORMATION

A. Differential Geometry Coordinates

In section II, we have proposed a modeling method for linear object deformation. This method can be applied to a sheet object if the shape of the object is regarded as rectangle, namely, the object has belt-like shape. However, in section II, it is assumed that the shape of cross-section of a linear object is fixed. This assumption is not appropriate to represent 3D shape of a belt object because the shape of its cross-section can change due to deformation. Therefore, in this section, we will adapt our method for linear object deformation to belt object deformation.

First, we formulate the deformation of a belt object in 3D space. Assumptions in this section are as follows:

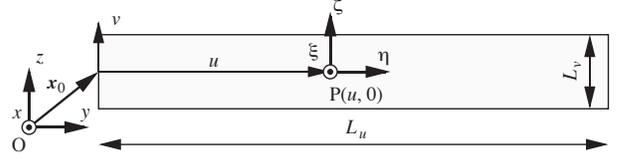


Fig. 7. Coordinate systems describing belt object deformation

- A belt object has rectangular shape.
- The width of the belt object is sufficiently small compared to its length.
- The object is inextensible. Namely, it can be bent and twisted but cannot be expanded or contracted.
- Both ends of the object cannot be deformed because connectors are attached to the ends.

We focus on deformation of the central axis in a longitudinal direction of a belt object and attempt to represent the whole shape of the object using it.

Let L_u and L_v be the length and the width of the object, respectively. Let u be the distance from one end of the object along the central axis in its longitudinal direction, corresponding to s in case of a linear object. Let v be the distance from the central axis in a transverse direction of the object. The global space coordinate system and the local object coordinate systems at individual points on the object are introduced as shown in Fig.7. Assume that the central axis in a longitudinal direction of the object is parallel to the y -axis and the normal vector of any point on the object is parallel to the x -axis in its natural state. Deformation of the object is then represented by use of Eulerian angles, $\phi(u, 0)$, $\theta(u, 0)$, and $\psi(u, 0)$. Let $\mathbf{x}(u, 0) = [x(u, 0), y(u, 0), z(u, 0)]^T$ be the position vector of point $P(u, 0)$. The position vector can be computed by integrating vector $\boldsymbol{\eta}(u, 0)$. Namely,

$$\mathbf{x}(u, 0) = \mathbf{x}_0 + \int_0^u \boldsymbol{\eta}(u, 0) du, \quad (31)$$

where $\mathbf{x}_0 = [x_0, y_0, z_0]^T$ is the position vector at the end point $P(0, 0)$.

Infinitesimal ratios of rotational angles ω_ξ , ω_η , and ω_ζ are also defined. In case of a linear object, ω_ξ and ω_η represents flexure of the object and ω_ζ corresponds to torsion of the object. In case of a belt object, ω_ζ and ω_η correspond to flexure and torsion of the object, respectively. Note that ω_ξ indicates curvature of the central axis of the object in uv -space.

B. Description of Surface Bending

Next, we consider general description of 3D surface. Let $\mathbf{x}(u, v)$ be the position vector of point $P(u, v)$ on a surface. Let $\mathbf{x}_u(u, v)$ and $\mathbf{x}_v(u, v)$ be tangent vectors at point $P(u, v)$ along u - and v -axes, respectively, and let $\mathbf{e}(u, v)$ be the normal vector at point $P(u, v)$. According to differential geometry, the normal curvature κ in direction $\mathbf{d} = a\mathbf{x}_u + b\mathbf{x}_v$ is represented as follows:

$$\kappa = \frac{La^2 + 2Mab + Nb^2}{Ea^2 + 2Fab + Gb^2}, \quad (32)$$

where E , F , and G are coefficients of the first fundamental form and L , M , and N are those of the second fundamental form of the surface. These coefficients are defined as follows:

$$E = \mathbf{x}_u \cdot \mathbf{x}_u, \quad F = \mathbf{x}_u \cdot \mathbf{x}_v, \quad G = \mathbf{x}_v \cdot \mathbf{x}_v, \quad (33)$$

$$L = \frac{\partial \mathbf{x}_u}{\partial u} \cdot \mathbf{e}, \quad M = \frac{\partial \mathbf{x}_u}{\partial v} \cdot \mathbf{e}, \quad N = \frac{\partial \mathbf{x}_v}{\partial v} \cdot \mathbf{e}. \quad (34)$$

The normal curvature κ depends on the direction \mathbf{d} and its maximum value κ_1 and its minimum value κ_2 are called the principal curvatures. Direction \mathbf{d}_1 of the maximum curvature κ_1 and direction \mathbf{d}_2 of the minimum curvature κ_2 are referred to as principal directions. The principal curvatures and the principal directions specify bend of a surface. A surface is also characterized by Gaussian curvature $K(u, v)$ and the mean curvature $H(u, v)$. They are related to the principal curvatures κ_1 and κ_2 by

$$K = \kappa_1 \kappa_2 = \frac{LN - M^2}{EG - F^2}, \quad (35)$$

$$H = \frac{\kappa_1 + \kappa_2}{2} = \frac{EN - 2FM + GL}{2(EG - F^2)}. \quad (36)$$

Vectors \mathbf{x}_u , \mathbf{x}_v , and \mathbf{e} correspond to $\boldsymbol{\eta}$, $\boldsymbol{\zeta}$, and $\boldsymbol{\xi}$ in this paper, respectively. Then, coefficients of the first fundamental form are $E = 1$, $F = 0$, and $G = 1$, respectively. Moreover, the derivation of unit vectors $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ can be described using infinitesimal ratios of rotational angles as follows:

$$\frac{\partial \boldsymbol{\eta}}{\partial u} = -\omega_\zeta \boldsymbol{\xi} + \omega_\xi \boldsymbol{\zeta}, \quad (37)$$

$$\frac{\partial \boldsymbol{\zeta}}{\partial u} = \omega_\eta \boldsymbol{\xi} - \omega_\xi \boldsymbol{\eta} = \frac{\partial \boldsymbol{\xi}}{\partial v}. \quad (38)$$

Substituting (37) and (38) into (34), L and M can be represented as a function of infinitesimal angle ratios as follows:

$$L = (-\omega_\zeta \boldsymbol{\xi} + \omega_\xi \boldsymbol{\zeta}) \cdot \boldsymbol{\xi} = -\omega_\zeta, \quad (39)$$

$$M = (\omega_\eta \boldsymbol{\xi} - \omega_\xi \boldsymbol{\eta}) \cdot \boldsymbol{\xi} = \omega_\eta. \quad (40)$$

In contrast, N cannot be described by Eulerian angles. So, we introduce the fourth parameter $\delta(u, 0)$: $N = \delta(u, 0)$. It corresponds to the curvature in a transverse direction. Consequently, Gaussian curvature K and the mean curvature H is described by

$$K = -\omega_\zeta \delta - \omega_\eta^2, \quad (41)$$

$$H = \frac{-\omega_\zeta + \delta}{2}. \quad (42)$$

Thus, bending of a surface is characterized by Eulerian angles $\phi(u, 0)$, $\theta(u, 0)$, and $\psi(u, 0)$ and the curvature in a transverse direction $\delta(u, 0)$. Note that K and H depends on not only coordinate u but also coordinate v . In this paper, we assume that the whole shape of a belt object can be described by the shape of the central axis in a longitudinal direction because the width of a belt object is sufficiently small compared to its length.

If a principal curvature κ_2 , *i.e.*, the minimum value of the normal curvature is equal to zero, the surface is developable. Namely, it can be flattened without its expansion or contraction. Such surface is referred to as a *developable surface*.

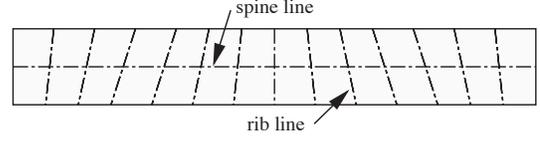


Fig. 8. Fishbone model

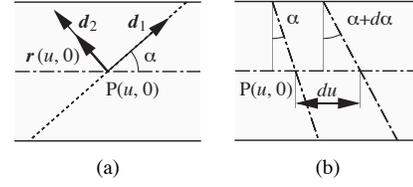


Fig. 9. Rib angle and rib lines

In this paper, we assume that a belt object is inextensible. Then, the deformed shape of the object corresponds to a developable surface. It means that the object is flexed in direction \mathbf{d}_1 and it is not deformed in direction \mathbf{d}_2 . Namely, a line the direction of which coincides with direction \mathbf{d}_2 is kept straight after deformation. In this paper, the central axis in a longitudinal direction of the object is referred to as the *spine line* and a line with zero curvature at a point on the object is referred to as a *rib line* as shown in Fig.8. We assume that flexure and torsion of the spine line and direction of the rib line of each point specifies deformation of a belt object. This model is referred to as a *fishbone model* in this paper. Let $\alpha(u, 0)$ be rib angle, which is the angle between the spine line and direction \mathbf{d}_1 as shown in Fig.9-(a). Let $\mathbf{r}(u)$ be a unit vector along a rib line at point $P(u, 0)$ on the spine line. It is described by

$$\mathbf{r} = -\boldsymbol{\eta} \sin \alpha + \boldsymbol{\zeta} \cos \alpha. \quad (43)$$

Then, coordinates of a point on a rib line and on either longitudinal edge $\mathbf{x}(u', \pm V/2)$ is represented as follows:

$$\mathbf{x}(u', \pm V/2) = \mathbf{x}(u, 0) \pm \frac{V}{2 \cos \alpha(u, 0)} \mathbf{r}(u, 0), \quad (44)$$

where u' satisfies

$$u' = u + \frac{V}{2} \tan \alpha(u, 0). \quad (45)$$

Consequently, the whole shape of a belt object can be represented using five variables $\phi(u)$, $\theta(u)$, $\psi(u)$, $\delta(u)$, and $\alpha(u)$. Note that they depend on only the distance u from one end of the object along the spine line.

C. Constraints on Belt Object Variables

Let us consider conditions which five variables must satisfy so that the surface of a belt object is developable. Gaussian curvature K of a developable surface must be zero at any point. So, the following constraint is imposed on the object.

$$K = -\omega_\zeta \delta - \omega_\eta^2 = 0, \quad \forall u \in [0, U]. \quad (46)$$

From (46), δ is described by

$$\delta = -\frac{\omega_\eta^2}{\omega_\zeta}. \quad (47)$$

Recall that infinitesimal ratio of rotational angle around ξ -axis ω_ξ indicates curvature of the spine line on the object. In the initial state, the spine line is straight, that is, its curvature is constantly equal to zero. So, ω_ξ must be satisfied the following equation after any deformation because of the inextensibility of a belt object:

$$\omega_\xi = 0, \quad \forall u \in [0, U]. \quad (48)$$

Moreover, as shown in Fig.9-(b), to prevent rib lines from intersecting with themselves on a belt object, the following inequalities must be satisfied:

$$\frac{V}{2} \tan \alpha + du \geq \frac{V}{2} \tan(\alpha + d\alpha), \quad (49)$$

$$\frac{V}{2} \tan(\alpha + d\alpha) + du \geq \frac{V}{2} \tan \alpha. \quad (50)$$

Then, rib angle α at any point on the spine line must be satisfied

$$-\frac{2 \cos^2 \alpha}{V} \leq \frac{d\alpha}{du} \leq \frac{2 \cos^2 \alpha}{V}, \quad \forall u \in [0, U]. \quad (51)$$

Substituting (47) into (32) and (42), The normal curvature in direction $\mathbf{d}_1 = \xi \cos \alpha + \eta \sin \alpha$, i.e., a principal curvature κ_1 is as follows:

$$\begin{aligned} \kappa_1 &= -\omega_\zeta \cos^2 \alpha + 2\omega_\eta \cos \alpha \sin \alpha - \frac{\omega_\eta^2}{\omega_\zeta} \sin^2 \alpha \\ &= -\omega_\zeta - \frac{\omega_\eta^2}{\omega_\zeta} \end{aligned} \quad (52)$$

Then, α can be described as follows:

$$\alpha = -\tan^{-1} \frac{\omega_\eta}{\omega_\zeta}. \quad (53)$$

Now, let us introduce parameter $\beta(u)$:

$$\beta = \tan \alpha. \quad (54)$$

Then, β must satisfy the following equation from (53):

$$\omega_\eta + \omega_\zeta \beta = 0, \quad \forall u \in [0, U]. \quad (55)$$

Moreover, (51) is described as follows by substituting (54):

$$-\frac{2}{V} \leq \frac{d\beta}{du} \leq \frac{2}{V}, \quad \forall u \in [0, U]. \quad (56)$$

Consequently, the shape of a belt object can be represented by four variables $\phi(u)$, $\theta(u)$, $\psi(u)$, and $\beta(u)$. And, they must satisfy (48), (55), and (56) in any state to maintain developability.

D. Potential Energy and Geometric Constraints

Let us formulate the potential energy of a deformed belt object. We can assume that a belt object is flexed along direction \mathbf{d}_1 without torsional deformation. This implies that the shape of cross-section along rib line is fixed while that along a transverse direction can change. Then, the potential energy U can be described as follows assuming that the flexural energy is proportional to the bending moment at each point $P(u)$:

$$U = \int_0^{L_u} \frac{R_f}{2} \kappa_1^2 du = \int_0^{L_u} \frac{R_f}{2} \frac{(\omega_\zeta^2 + \omega_\eta^2)^2}{\omega_\zeta^2} du, \quad (57)$$

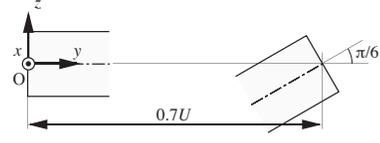


Fig. 10. Example of belt object deformation

where R_f represents the flexural rigidity of a belt object along the spine line at point $P(u)$.

Geometric constraints imposed on a belt object can be represented as described by (12) through (14). Therefore, the shape of a belt object is determined by minimizing the potential energy described by (57) under necessary constraints for developability described by (48), (55), and (56) and geometric constraints imposed on the object described by (12) through (14).

VI. VERIFICATION OF BELT OBJECT MODEL

A. Computation of Belt Object Deformation

In this section, numerical examples demonstrate how the proposed method computes the deformed shape of a belt object. The deformed shape of a belt object also can be computed by use of the algorithm proposed in section II. Let us express functions $\phi(u)$, $\theta(u)$, $\psi(u)$, and $\beta(u)$ by linear combinations of basic functions $e_1(u)$ through $e_n(u)$ as described by (16) through (19). Then, the deformed shape of a belt object can be derived by computing a set of coefficient vectors \mathbf{a}^ϕ , \mathbf{a}^θ , \mathbf{a}^ψ , and \mathbf{a}^β . In the following examples, basic functions described by (20) are used. Suppose that the length of the object L_u is equal to 1, its width L_v is equal to 0.1, and its flexural rigidity along the spine line R_f is constantly equal to 1. Necessary constraints for developability described by (48), (55), and (56) are divided into 16 conditions at point $P(iL_u/15)$ ($i = 0, \dots, 15$) respectively in the following examples.

Fig.10 shows an example of belt object deformation. In this example, positional and orientational constraints are described by

$$\mathbf{x}(L_u) = \int_0^{L_u} \boldsymbol{\eta}(u) du = \begin{bmatrix} 0 \\ 0.7 \\ 0 \end{bmatrix} L_u, \quad (58)$$

$$\phi(0) = \theta(0) = \psi(0) = \beta(0) = 0, \quad (59)$$

$$\boldsymbol{\eta}(L_u) = \begin{bmatrix} 0 \\ \cos(\pi/6) \\ \sin(\pi/6) \end{bmatrix}, \quad \boldsymbol{\zeta}(L_u) = \begin{bmatrix} 0 \\ -\sin(\pi/6) \\ \cos(\pi/6) \end{bmatrix} \quad (60)$$

$$\beta(L_u) = 0. \quad (61)$$

Namely, both end of the spine line are on the same line but directions of the spine line at these points are different. Fig.11 shows computational results. Fig.11-(a), -(b), and -(c) illustrate the top, front, and side view of the object, respectively. As shown in this figure, the object is flexed and twisted to satisfy given geometric constraints. Coordinates of the highest point of the object are $[0.3L_u, 0.4L_u, -0.01L_u]^T$.

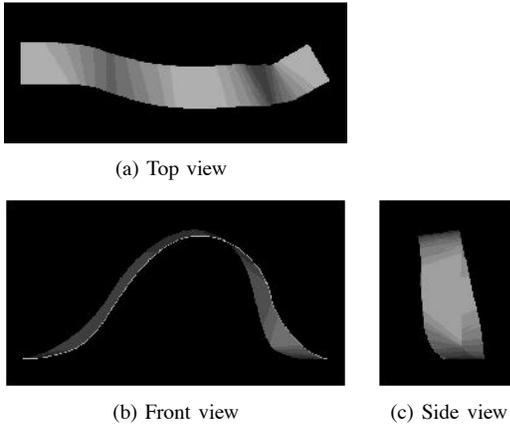


Fig. 11. Computational result of belt object deformation

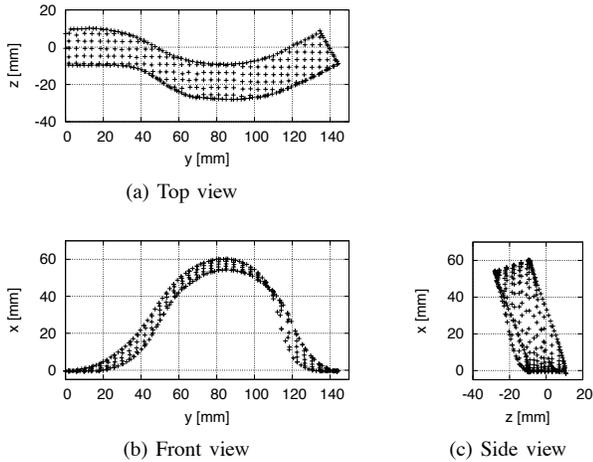


Fig. 12. Experimental result of belt object deformation

B. Measurement of Belt Object Deformation

Next, the computation result is experimentally verified by measuring the deformed shape of a belt object. We measured the shape of a rectangular polystyrene sheet which is 200mm long, 20mm wide, and 140 μ m thick with a 3D scanner. Their flexural rigidity is unknown but from (57), it is found that the deformed shape is independent of it when it is constant along the spine line. Fig.12 shows the experimental result of deformation illustrated in Fig.10. As shown in this figure, the computed shape on xy - and xz -planes is qualitatively similar to the actual shape and x - and y -coordinates of the object peak almost coincide. Thus, our method can estimate flexural and torsional deformation of a rectangular belt object using only flexural rigidity of the object along its spine line if the object is isotropic.

C. Discussion of Fishbone Model

In this section, we discuss the fishbone model. Recall that the surface of an inextensible belt object corresponds to a developable surface, which is a kind of ruled surfaces. A ruled surface is a surface that can be swept out by moving a straight line, which is called a ruling, in 3D space and it can be formulated as follows:

$$\mathbf{x}(u, v) = \mathbf{p}(u) + v\mathbf{q}(u), \quad (62)$$

where $\mathbf{p}(u)$ and $\mathbf{q}(u)$ are referred to as the base curve and the director curve, respectively. Rib lines in the fishbone model correspond to rulings. Moreover, \mathbf{x} and \mathbf{r} in (44) are similar to the base curve and the director curve, respectively. The formulation described by (62) is sufficient to represent the object surface after deformation. However, it is not suitable for representation of energy increment from the initial shape. To estimate potential energy of the object and to derive its stable shape, we have to specify dependent parameters on deformation and independent parameters of deformation. As a belt object is assumed to be inextensible, its shape in uv -space is not changed by any deformation. This means that the length, width, and angle between u - and v -axes are constant. So, $E = 1$, $G = 0$, $F = 1$. Furthermore, the constraint described by (48) is added for straightness of the spine line in uv -space. Then, the object only can be flexed around ζ -axis and twisted around η -axis, and the rib angle is determined from these flexure and torsion. As mentioned before, the object shape is represented by four variables $\phi(u)$, $\theta(u)$, $\psi(u)$, and $\beta(u)$. Note that they must satisfy constraints described by (48) and (55). Therefore, we can conclude that deformation of an inextensible belt object is described by two independent variables.

D. Application of Fishbone Model

Some flexible circuit boards bend like a polygonal line or curve like a circular arc. Let us discuss application of our model to such bent/curved boards. First, to represent a belt object with multiple bends, Eulerian angles and rib angles of straight parts between bends should be defined separately. The deformed shape of the object is then derived by minimizing total potential energy of each part. But, continuity of the rib line at each bend should be discussed. Fig.14 shows a computational result of deformation of a belt object with one bend illustrated in Fig.13. Next, let us consider a curved belt object. As we assume that the spine line is straight in this paper, ω_ξ is constantly equal to zero. If an object is curved with a certain curvature, ω_ξ must be equal to that curvature even if the object deforms. We can impose this constraint on the object instead of (48). This implies that our proposed method can be applied to a curved belt object. Fig.16 shows a computational result of deformation of a curved belt object illustrated in Fig.15. Thus, our proposed method can represent deformation of various belt objects.

VII. CONCLUSIONS

In this paper, a modeling method of linear/belt object deformation based on differential geometry was proposed. First, differential geometry was extended to describe linear object deformation including flexure, torsion, and extension. The shape of a linear object can be described by four independent variables if extensible and by three otherwise. Next, the validity of this method was verified with measuring experiments. Moreover, it was shown that more complex shapes such as knots and knitted fabrics also can be computed using our proposed approach. Finally, this approach was applied to deformation of an inextensible belt object. It

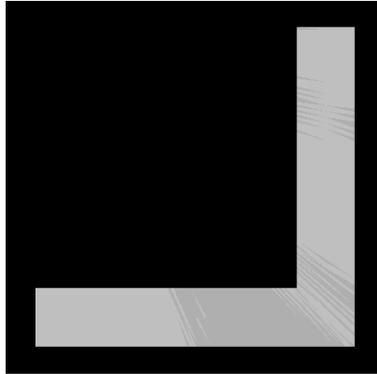
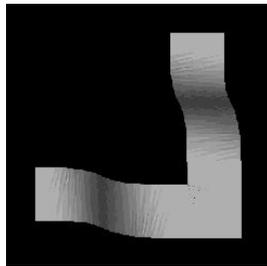
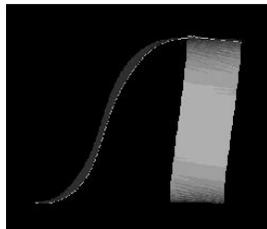


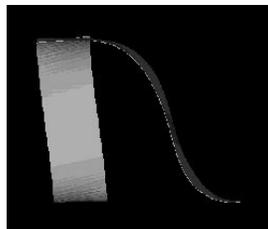
Fig. 13. Bent Belt Object



(a) Top view



(b) Front view



(c) Side view

Fig. 14. Deformation of Bent Belt Object

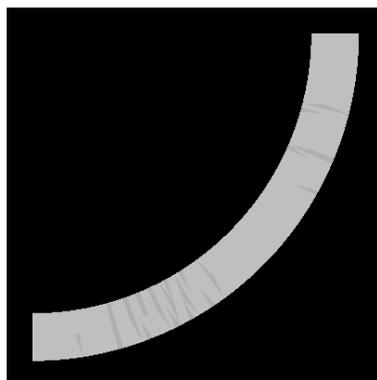
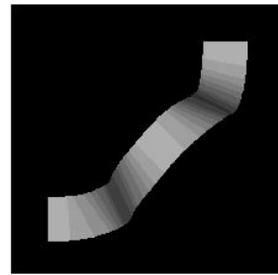
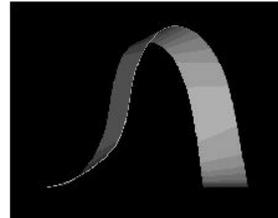


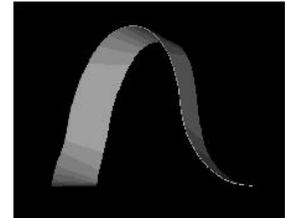
Fig. 15. Curved Belt Object



(a) Top view



(b) Front view



(c) Side view

Fig. 16. Deformation of Curved Belt Object

was found that the belt object shape can be described by two independent variables.

REFERENCES

- [1] S. Timoshenko, *Strength of Materials Part I Elementary Theory and Problems*, D. Van Nostrand Company, Inc., 1955.
- [2] T. Belytschko, W. K. Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures*, John Wiley & Sons, 2000, pp.509–568.
- [3] S. Timoshenko, *Theory of Plates and Shells*, McGraw-Hill Book Company, Inc., 1940.
- [4] E. H. Mansfield, "The Inextensional Theory for Thin Flat Plates", *The Quarterly J. Mechanics and Applied Mathematics*, Vol.8, 1955, pp.338–352.
- [5] H. M. Irvine, *Cable Structures*, MIT Press, 1981.
- [6] J. Weil, "The Synthesis of Cloth Objects", *Computer Graphics*, Vol.20, No.4, 1986, pp.49–54.
- [7] J. W. Eischen, S. Deng, and T. G. Clapp, "Finite-Element Modeling and Control of Flexible Fabric Parts", *IEEE Computer Graphics and Applications*, 1996, pp.71–80.
- [8] B. Eberhardt, A. Weber, and W. Strasswer, "A Fast, Flexible, Particle-System Model for Cloth Draping", *IEEE Computer Graphics and Applications*, 1996, pp.52–59.
- [9] D. Baraff and A. Witkin, "Large Steps in Cloth Simulation", *Computer Graphics Proceedings (SIGGRAPH 1998)*, 1998, pp.43–54.
- [10] D. K. Pai, "STRANDS: Interactive Simulation of Thin Solids using Cosserat Models", *Computer Graphics Forum*, Vol.21, No.3, 2002, pp.347–352.
- [11] A. Gray, *Modern Differential Geometry of Curves and Surfaces*, CRC Press, 1993.
- [12] M. Moll and L. E. Kavraki, "Path Planning for Variable Resolution Minimal-Energy Curves of Constant Length", *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp.2142–2147.
- [13] H. Goldstein, *Classical Mechanics*, Addison-Wesley, 1980.
- [14] L. E. Elsgolc, *Calculus of Variations*, Pergamon Press, 1961.
- [15] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, 1976.
- [16] J. Phillips, A. Ladd, and L. E. Kavraki, "Simulated Knot Tying", *Proc. IEEE Int. Conf. Robotics and Automation*, 2002, pp.841–846.
- [17] H. Shi and S. Payandeh, "Real-Time Knotting and Unknotting", *Proc. IEEE Int. Conf. Robotics and Automation*, 2007, pp.2570–2575.
- [18] B. G. A. V. Leaf, "Models of the Plain-Knitted Loop", *J. Textile Institute Transactions*, 1960, pp.49–58.
- [19] T. Wada, S. Hirai, T. Hirano, and S. Kawamura, "Modeling of Plain Knitted Fabrics for Their Deformation Control", *Proc. IEEE Int. Conf. Robotics and Automation*, 1997, pp.1960–1965.

Rope Knotting and Unknotting with Haptic Feedback

Hans Fuhan Shi and Shahram Payandeh

Abstract—In this paper, we present a mechanics-based approach to simulation of deformable linear objects (DLOs) with visual and force feedback. In our rope model, which can represent the mechanical properties of a real thread such as stretching, compressing, bending, and twisting, we simulate not only external forces, but also internal forces including the friction force during knotting and unknotting. We also present how forces propagate along the rope when the user pulls it with one or two hands. We developed a simulator to allow users to grasp and smoothly manipulate a virtual rope, and to tie an arbitrary knot.

I. INTRODUCTION

The application of knots can ascend to the Paleolithic era. We use all kinds knots in our everyday lives such as fastening our shoes or clothes, wrapping gifts, animal handling, fishing, sailing, climbing, carving, and even for decoration etc. In the medical field, they are essential to suturing in today’s surgery procedures. Real-time simulation of deformable linear objects (DLOs) is need in many areas, such as surgical training systems and rock climbing or sailing training systems to teach users how to tie and un-tie a knot. It also related to cloth-like deformable objects simulation, an area has attracted much attention in Computer Graphics recently.

Computer-based training systems, using computers and electromechanical user interface devices, open new possibilities in training, offering many benefits compared to traditional training methods. Real-time knotting and unknotting simulations raise unique and difficult issues because of the rope’s deformability, difficulty of collision detection and management, and the demanding requirements of force feedback output. In this paper, we developed a simulator in our training environment to allow users to tie and untie any kind of knot.

There are a number of works which have made some contributions to the development of DLO simulation. Most of these previous models can be categorized as geometry-based models or mechanics-based models. Geometric models are slightly less accurate because they only simulate displacements. Mechancis-based models are often more accurate, although, for the virtual reality simulation, they may shift continuously until converge to equilibrium points, which makes them difficult for users to manipulate. In our training systems, because the purpose is to enable users to feel the

force feedback when they manipulate the rope, especially during knotting and unknotting, to make it more realistic, we need to consider both external and internal forces to determine the force output. Thus geometric models are obviously inappropriate.

Some researchers have been focusing on knotting manipulation by robots. In [1], Wakamatsu, Arai and Hirai established a model of DLOs based on an extension of differential geometry, and proposed a planning method for knotting/unknotting of DLOs based on the knot theory. If the initial and the objective states of the linear object are given, all possible knotting/unknotting plans can be derived and be executed by their system. However, their proposed models can not simulate the DLOs dynamically in 3D space. In addition, their system does not allow any user interaction, and can not simulate the knotting/unknotting procedure in real-time. [2] describes a 2D DLOs dynamic model based on the differential geometry coordinates. In [3], a knot planning from observation(KPO) system is described. First, this system observes the procedure of tying a knot by a human as a sequence of movement primitives. Then, by repeating the sequence, it can tie a similar knot. The topological information of a knot is represented in a P-data representation. In [4], a topological motion planner for manipulating DLOs and tying knots using cooperating robot arms was introduced based on Probabilistic RoadMaps (PRMs). Our initial system is to allow interaction where the user can form and tie a knot. In the future work, we can include a real-time planner which can assist and guide the user during the knotting task. However, for the virtual reality simulation, we must allow a user to interact with the computer-simulated environment, and enable the user to feel, touch, and manipulate the virtual objects such as during virtual suturing tasks.

In [5] [6] [7], a Cosserat approach of modeling DLOs based on the Cosserat theory of elastic rods has been introduced. Cosserat model is well suited for real-time applications because it needs less computation compared to finite elements models and provides a clear delineation between basic physical principles, material properties and mathematical approximations. However, in return, it yields a set of ordinary differential equations to be solved. If two end points or multiple points along the length of a suture are specified (as in the procedure of knotting or unknotting with two hands), it is significantly more difficult to solve these equations. In addition, the “shooting” technique which is mentioned in [5] makes it very difficult to integrate external forces [8].

A particle-based model of a rope is represented in [9]

H. Shi is with Experimental Robotics Laboratory, School of Engineering Science, Simon Fraser University, Burnaby, BC V5A 1S6, CANADA fuhans@ensc.sfu.ca

S. Payandeh is with Experimental Robotics Laboratory, School of Engineering Science, Simon Fraser University, Burnaby, BC V5A 1S6, CANADA shahram@ensc.sfu.ca

by overlapping spheres representing mass-points, which are connected by simple springs. Each mass-point can collide with other mass points as in the instantaneous elastic collision model, but the author only considers the linear spring forces and does not allow any user interaction. In [10], inner bending force and the gravity are taken into consideration. In [11], the author mentioned gravity, stretch/compression force, forces from bending and twisting, dissipative friction, and contact forces with environment or to self-collision, but there is no detail about how to compute those forces.

A mass-spring model for suture in surgical training system has been built in [12]. Torsional spring, torsional damper, and viscous damper are mentioned in this paper, but, the author did not use them in the simulation due to the complex computation. Further more, there is no discussion about collision detection and force propagation for haptic interaction between the user and the suture model.

Our rope model is built based on all the force definition given in [12], and we provide a user-interface to allow users tie an arbitrary knot. Also we analyze how the forces propagate along the suture during knotting and unknotting. With the virtual coupling technique[13], we can provide very smooth force feedback to the user.

The rest of this paper is organized as follows: Section II describes the model we are using and illustrates how to calculate internal and external forces. Section III covers how forces propagate along the suture during knotting and unknotting. Section IV describes case studies and results, and section V gives the conclusion and discusses about the future work.

II. MODEL DESCRIPTION

For 1D element, we model our rope as a mass-spring system which consists of a sequence of mass points laying on the centreline of the suture. (see (a) of Fig. 1).

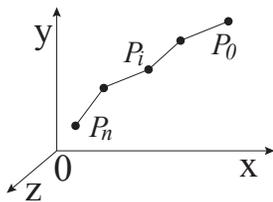


Fig. 1. Rope model

During graphic rendering, we use cylinders as rope segments connecting two successive points. We use Euler method to calculate the shape of our rope. First we compute the total force acting at each point, P_i , and then update its position based on the computed force. Once the total force at each of the nodes has been calculated, with the interval time dt , we can obtain the velocity and position of each point.

The following part of this section explains the forces we simulate in our simulator. We can use various combinations of these forces to build different models. The springs and dampers both contribute some force to the net force \mathbf{f} at each

point. Different springs and dampers all behave differently and we calculate their force contributions using their own particular equations.

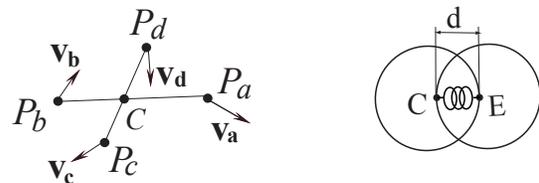
A. External Forces

The external forces include the gravitational force, the user input forces through haptic devices, the friction forces during knotting or unknotting, as well as the contact force with obstacles:

1) *Gravity*: $\mathbf{f}_g = Gm$. where $G = 9.8N/kg$, and m is the mass of the mass point.

2) *User Input Force*: Allowing the user to provide both input and output to the simulation in the form of forces, positions, and velocity etc, a haptic device becomes a natural interface for a dynamic simulation, which needs to calculate all the forces applied to the objects. However, a position controlled impedance style haptic device, such as PHANTOM Omni and PHANTOM Desktop from Sensable Tchenology, forces are not directly available as input variables into the model. Furthermore, the mechanical characterization and digital nature of the haptic device make the operation of directly incorporating the device as part of the simulation more challenging. To overcome these difficulties, we use virtual coupling technique which introduces a indirect layer of interaction between the mechanical device and the simulation by employing a spring-damper between a simulated body and the device end-effector Another advantage of this technique is that we can use different constants for computing the output force for the device versus the input force for the simulated body, which makes the forces appropriate for both the haptic device and the dynamic simulation.

3) *Friction Force*: In this paper, we only consider dynamic friction forces, like sliding friction, during the procedure of knotting and unknotting, and we do not consider any static frictions. During the simulation, we use Coulomb's model and consider each rope segment as rigid body, hence we can not bend to any angle for any instant time. From Coulomb's observations we know that: kinetic frictional force is approximately independent of contact area and velocity magnitude of the object; Coefficient of friction depends on pairs of materials. During knotting or unknotting procedure, suppose there are only two segments colliding with each other (see Fig. 2). Let μ be the friction constant,



(a) Two Sliding Segments (b)Two Segments Intersection

Fig. 2. Friction during knotting / unknotting

$\hat{\mathbf{e}}$ be the friction direction vector, \mathbf{n} be the force of repulsion,

then the friction \mathbf{f}_f can be described as:

$$\mathbf{f}_f = \mu \|\mathbf{n}\| \hat{\mathbf{e}}. \quad (1)$$

To calculate the repulsion force \mathbf{n} , we introduce a spring-damper between the contact point C and the end point E .

$$\begin{aligned} \mathbf{n} &= (k_{rs}d - k_{rd}(\mathbf{v}_r \cdot \hat{\mathbf{n}}))\hat{\mathbf{n}}, \\ d &= 2r - s. \end{aligned} \quad (2)$$

where k_{rs} is a spring constant for the repulsion force, r is the radius of the rope model, d is the distance between contact point C to point E (see (b) of Fig. 2), s is the distance between the two centers of the contact cylinder, k_{rd} is the damper constant for the repulsion force, \mathbf{v}_r is the relative velocity of point C with respect to point E , $\hat{\mathbf{n}}$ is the unit vector from point E to point C . We use linear interpolation to compute the velocity of a point on the segment. For example (see Fig. 2), $\mathbf{v}_c = (1-a)\mathbf{v}_a + a\mathbf{v}_b$, where a is the fraction of point C along P_aP_b , $\mathbf{v}_e = (1-b)\mathbf{v}_c + b\mathbf{v}_d$, where b is the fraction of point E along P_cP_d . Then the relative velocity $\mathbf{v}_r = \mathbf{v}_c - \mathbf{v}_e$, and the friction direction vector $\hat{\mathbf{e}}$ is computed as follows:

$$\hat{\mathbf{e}} = \frac{(\mathbf{v}_r \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} - \mathbf{v}_r}{\|(\mathbf{v}_r \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} - \mathbf{v}_r\|}. \quad (4)$$

B. Internal Forces

1) *Linear spring force*: The linear spring force is computed by comparing the current segment length, l_i , between point, P_i and P_{i+1} , with the rest length of the segment l_r , and by projecting the resulting difference on the direction from point P_i to P_{i+1} . Then, $l_i = \|P_{i+1} - P_i\|$, $\Delta l = \frac{l_i - l_r}{l_r}$, and l_r is the rest length between point, P_i and P_{i+1} . Let $\hat{\mathbf{e}}_i$ be the unit vector from point, P_i to P_{i+1} , then,

$$\hat{\mathbf{e}}_i = \frac{P_{i+1} - P_i}{\|P_{i+1} - P_i\|}, \quad (5)$$

$$\mathbf{f}_s = k_l \Delta l \hat{\mathbf{e}}_i. \quad (6)$$

where k_l is the linear spring constant.

2) *Linear damper*: We simulate all the factors that try to stop the spring as it moves as one constant called the damping factor, k_d . This force opposes the direction of movement and is proportional to the velocity of the moving mass. When the system is at rest ($\mathbf{v} = 0$), no linear damping force is involved.

$$\mathbf{f}_d = k_d(v_{i+1} - v_i)\hat{\mathbf{e}}_i. \quad (7)$$

where $v_{i+1} = \mathbf{v}_{i+1} \cdot \hat{\mathbf{e}}_i$, $v_i = \mathbf{v}_i \cdot \hat{\mathbf{e}}_i$, k_d is the linear damper constant. v_{i+1} and v_i are the norms of the components of the velocity of point P_{i+1} and P_i on the direction $\hat{\mathbf{e}}_i$.

3) *Torsional spring*: The torsional spring is derived from the angle, α , between two connected segments of the rope. The basic idea is to model each two connected segments as a triangle with a spring as the hypothesis pushing the end points to the full expanded position. The length of the two connected segments remain unchanged. Only the force component orthogonal to the segments is used for the end points (See (a) of Fig. 3). Let $\hat{\mathbf{e}}_{i-1}$ and $\hat{\mathbf{e}}_i$ be the unit

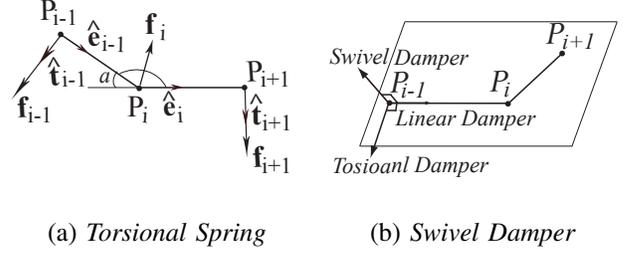


Fig. 3. Torsional Spring and Swivel Damper

vectors with directions from point, P_{i-1} to P_i , and from P_i to P_{i+1} , respectively. Let $\hat{\mathbf{t}}_{i-1}$ and $\hat{\mathbf{t}}_{i+1}$ be the unit vectors with directions the same as the torsional force applied at the two endpoints and therefore, orthogonal to $\hat{\mathbf{e}}_{i-1}$ and $\hat{\mathbf{e}}_i$ respectively. Then, $\hat{\mathbf{t}}_{i+1} = \hat{\mathbf{e}}_i \times (\hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i)$, $\hat{\mathbf{t}}_{i-1} = \hat{\mathbf{e}}_{i-1} \times (\hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i)$. If $\hat{\mathbf{e}}_{i-1} \cdot \hat{\mathbf{e}}_i \geq 0$, $\alpha = \arcsin(\|\hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i\|)$. If $\hat{\mathbf{e}}_{i-1} \cdot \hat{\mathbf{e}}_i < 0$, $\alpha = \pi - \arcsin(\|\hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i\|)$. The torsional spring force can be computed as follows:

$$\mathbf{f}_{i-1} = k_{ts} \frac{\alpha}{\pi \|P_{i-1} - P_i\|} \hat{\mathbf{t}}_{i-1}, \quad (8)$$

$$\mathbf{f}_{i+1} = k_{ts} \frac{\alpha}{\pi \|P_{i+1} - P_i\|} \hat{\mathbf{t}}_{i+1}, \quad (9)$$

$$\mathbf{f}_i = -(\mathbf{f}_{i-1} + \mathbf{f}_{i+1}). \quad (10)$$

where k_{ts} is the torsional spring constant.

4) *Torsional damper*: The torsional damper works against the torsional spring to prevent any harmonic motion from accumulating. Similar to the linear damper, it also models the internal friction that resists bending in regular objects. Let, v_{i-1} , v_{ib} , be the norms of the velocity components of, \mathbf{v}_{i-1} , and, \mathbf{v}_i , on the direction of, $\hat{\mathbf{t}}_{i-1}$, and let, v_{i+1} , v_{ia} , be the norms of the velocity components of, \mathbf{v}_{i+1} , and, \mathbf{v}_i , on the direction of, $\hat{\mathbf{t}}_{i+1}$. Then, the torsional damper on the points, P_{i-1} , P_i and P_{i+1} , can be computed by:

$$\mathbf{f}_{i-1} = \left(\frac{(v_{i-1} - v_{ib})}{\|P_{i-1} - P_i\|} + \frac{(v_{i+1} - v_{ia})}{\|P_{i+1} - P_i\|} \right) \frac{k_{td} \hat{\mathbf{t}}_{i-1}}{\|P_{i-1} - P_i\|}, \quad (11)$$

$$\mathbf{f}_{i+1} = \left(\frac{(v_{i-1} - v_{ib})}{\|P_{i-1} - P_i\|} + \frac{(v_{i+1} - v_{ia})}{\|P_{i+1} - P_i\|} \right) \frac{k_{td} \hat{\mathbf{t}}_{i+1}}{\|P_{i+1} - P_i\|}, \quad (12)$$

$$\mathbf{f}_i = -(\mathbf{f}_{i-1} + \mathbf{f}_{i+1}). \quad (13)$$

where k_{td} is torsional damper constant, $v_{i-1} = \mathbf{v}_{i-1} \cdot \hat{\mathbf{t}}_{i-1}$, $v_{ib} = \mathbf{v}_i \cdot \hat{\mathbf{t}}_{i-1}$, $v_{i+1} = \mathbf{v}_{i+1} \cdot \hat{\mathbf{t}}_{i+1}$, $v_{ia} = \mathbf{v}_i \cdot \hat{\mathbf{t}}_{i+1}$.

5) *Swivel damper*: Point, P_{i-1} , has a velocity relative to the center point, P_i . So far, two components of that relative velocity have been damped. There still remains a component perpendicular to those two. Without the dampening, point P_{i-1} could indefinitely orbit the line formed by extending the edge connecting point P_{i+1} and point P_i (See (b) of Fig. 3).

Let $\hat{\mathbf{s}}$ be the unit vector of the swivel dampers of point P_{i-1} and P_{i+1} , then, $\hat{\mathbf{s}} = \hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i$. The swivel dampers can

be computed by:

$$\mathbf{f}_{i-1} = k_{sw} \frac{(\mathbf{v}_{i-1} - \mathbf{v}_i) \cdot \hat{\mathbf{s}}}{\|P_{i-1} - P_i\|} \hat{\mathbf{s}}, \quad (14)$$

$$\mathbf{f}_{i+1} = k_{sw} \frac{(\mathbf{v}_{i+1} - \mathbf{v}_i) \cdot \hat{\mathbf{s}}}{\|P_{i+1} - P_i\|} \hat{\mathbf{s}}, \quad (15)$$

$$\mathbf{f}_i = -(\mathbf{f}_{i-1} + \mathbf{f}_{i+1}). \quad (16)$$

where k_{sw} is the swivel damper constant.

III. FORCE PROPAGATION ALONG THE ROPE

To prevent our rope from being stretched too long or compressed too short, we set l_{max} and l_{min} as the maximum and minimum length of one rope segment respectively. Let l_i be the segment length between P_i and P_{i+1} . To analyze the force propagation when the user grabs the rope, we need to compute the forces acting at each point from the grabbed point to the start point and to the end point of the rope. We define different scenarios as follows:

A. Condition A

Assume the user grabs point P_{i+1} with one hand. If $l_{min} < l_i < l_{max}$. There is no propagation of the user input force \mathbf{f}_h from point P_{i+1} to P_i . All the user input force has been converted to the internal forces along the rope.

B. Condition B

Assume the user grabs point P_{i+1} with one hand. If the expected segment length $l'_i > l_{max}$ or $l'_i < l_{min}$, we need to adjust the segment length to l_{max} or l_{min} (see (a) of Fig. 4).

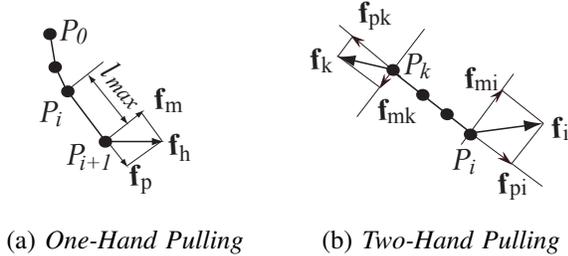


Fig. 4. Force Propagation

Let \mathbf{f}_p be the component of the input force \mathbf{f}_h along the segment direction, and \mathbf{f}_p is the input force propagated to point P_i from point P_{i+1} . \mathbf{f}_p and \mathbf{f}_m can be obtain from the following equations:

$$\mathbf{f}_p = (\mathbf{f}_h \cdot \hat{\mathbf{e}}_i) \hat{\mathbf{e}}_i, \quad (17)$$

$$\mathbf{f}_m = (\mathbf{f}_h \cdot \hat{\mathbf{e}}_m) \hat{\mathbf{e}}_m. \quad (18)$$

where $\hat{\mathbf{e}}_m = \frac{\hat{\mathbf{e}}_i \times \mathbf{f}_h}{\|\hat{\mathbf{e}}_i \times \mathbf{f}_h\|} \times \hat{\mathbf{e}}_i$, $\hat{\mathbf{e}}_i$ can be obtain from equation (5). Using the same method as above, we can derive the user input force propagated at each point of the rope.

C. Condition C

In this condition, we assume the user is pulling two points, P_k and P_i , of the rope. The method is almost the same as in condition B. But we need to do the propagation computation twice, first starting from point P_i , and then starting from point P_k (see (b) of Fig. 4).

IV. COLLISION DETECTION AND MANAGEMENT

First, we build a bounding-volume hierarchy (BVH) from the bottom-up representing the shape of the rope at successive levels of detail. (see Fig. 5).

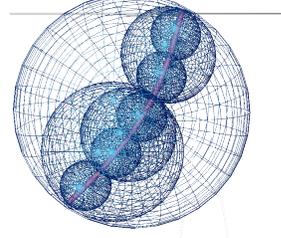


Fig. 5. Bounding-Volume Hierarchy

This method is similar to the method proposed in [16]. To find the self-collisions of the rope, we explore two copies of the BVH from the top down. Whenever two BVHs (one from each copy) are found to not overlap, we know that they cannot contain colliding segments, and hence, we do not explore their contents. When two leaf spheres overlap, the distance between the two centers of the nodes is computed. If it is less than the node diameter, $2r$, then the two segments are reported to collide. However, no node is ever considered to be in collision with itself or its immediate neighbors along the rope chain.

To find the collisions between the rope and grippers, we consider the gripper as a triangle between two jaws which are line segments with a given radius, and check if the BHV of the rope has any overlap with this triangle. If intersection happened, compute the intersection point which will be the grab point.

For self-collision of the rope, when two rope segments are detected to be at a distance $d < 2r$ from each other, then, an equal (but opposite) displacement vector is applied to each segment along. This displacement is just long enough to take the segments out of collision, with a slight "safety margin". Hence, each node is shifted away by $r - d/2 + \epsilon/2$ (see Fig. 6).

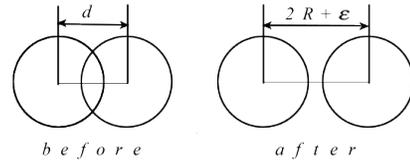


Fig. 6. Collision management of two rope segments

If a collision occurred, during real time simulation, we need to compute new velocities of mass points which are involved in the collision. Similarly to the method presented in [17], we apply impulses to the end points of these two segments. See Fig. 2 for the case where point C with relative

position a along the segment $\overrightarrow{P_a P_b}$ interacts with point E with relative position b along the segment $\overrightarrow{P_c P_d}$. Let \mathbf{i} be the impulse, then, $\mathbf{i} = \mathbf{n}\Delta t$, where \mathbf{n} is the repulsion force that we can obtain from equation (3). Then we can compute the new velocities as follows:

$$i' = \frac{2\|\mathbf{i}\|}{(a^2 + b^2 + (1-a)^2 + (1-b)^2)}, \quad (19)$$

$$\mathbf{v}_a^{new} = \mathbf{v}_a + (1-a)\frac{i'}{m}\hat{\mathbf{n}}, \quad (20)$$

$$\mathbf{v}_b^{new} = \mathbf{v}_b + a\frac{i'}{m}\hat{\mathbf{n}}, \quad (21)$$

$$\mathbf{v}_c^{new} = \mathbf{v}_c - (1-b)\frac{i'}{m}\hat{\mathbf{n}}, \quad (22)$$

$$\mathbf{v}_d^{new} = \mathbf{v}_d - b\frac{i'}{m}\hat{\mathbf{n}}. \quad (23)$$

where m is the mass of each mass point $P_a, P_b, P_c,$ and P_d . $\hat{\mathbf{n}}$ is the unit vector from point E to point C .

V. HAPTIC FORCE FEEDBACK

In the previous section, we have mentioned to calculate the user input force by introducing virtual coupling technique (see Fig. 7).

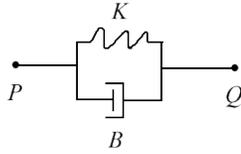


Fig. 7. Virtual coupling

Change the direction of the input force to the opposite, we can get the out put force which we need to feed the haptic device. Point P is the real position of the end factor and point Q is the grabbed point. By employing a spring-damper between a simulated body and the device end-effector, we can make the out put force as smooth as possible. We can also adjust the K and B to satisfy the out put requirements for different haptic devices.

In order to study the details inside our rope model, we took the 15th node as an example and plot the spring force, spring damper, torsional spring, torsional damper, and swivel damper acting on it when the rope swings freely (see Fig.8 - Fig. 13).

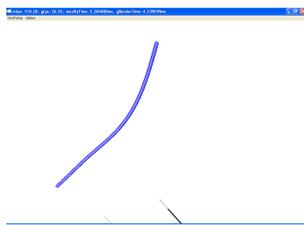


Fig. 8. Screen shot of the rope swings freely

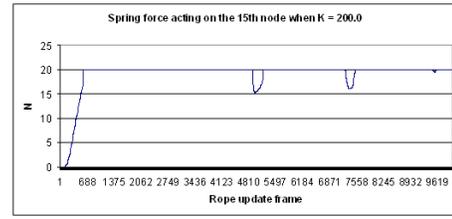


Fig. 9. Spring force acting on the 15th node

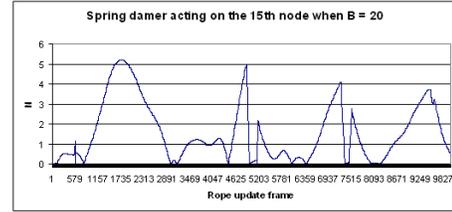


Fig. 10. Spring damper acting on the 15th node

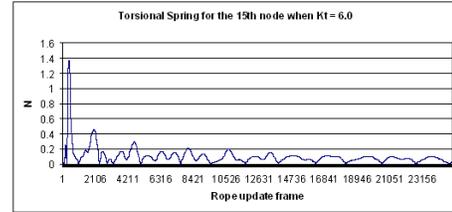


Fig. 11. Torsional spring acting on the 15th node

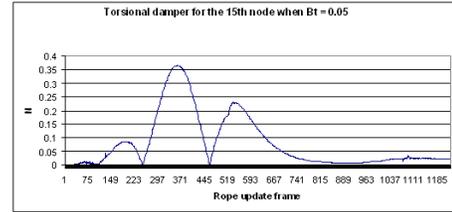


Fig. 12. Torsional damper acting on the 15th node

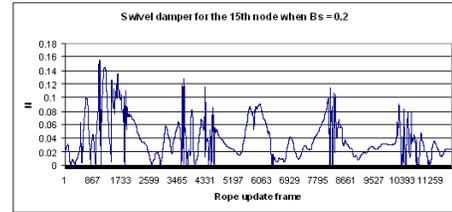


Fig. 13. Swivel damper acting on the 15th node

To demonstrate how friction force changes when the friction constant is changed, we plot the friction forces when the rope is colliding itself and changed the friction constant (see Fig. 14). Fig.15 to Fig. 18 are the friction force plots when $\mu = 0.1, 0.5, 1.0, 2.0$

Because the maximum exertable force for PHANTOM Omni is 0.75lbf (3.3N), we can not output the forces to the Haptic devices from virtual coupling spring directly. Therefore, we chose a constant equal to 0.003 to scale the forces before we feed them to PHANTOM Omni. We plot the forces which we send to PHANTOM Omni during each haptic update frame for one-hand pulling and two-hand

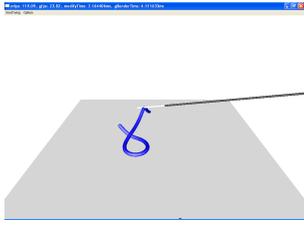


Fig. 14. Screen shot of rope colliding

knotting cases. Taking the magnitudes of the forces as y-axis and each haptic update frame as x-axis, we obtain the forces plots as in Fig. 19 to Fig 23

VI. CASE STUDY AND RESULT

A. Experiment Setup

Our simulation was implemented on a PC with dual 3.2G Intel®Pentium®4 CPUs and 512 MB memory. For physics-based models, the most challenging part is how to determine its parameters. If parameters are inappropriate, it may impact the whole system's stability or even over its limits. After many experiments, we chose our rope parameters as in Table I:

TABLE I
ROPE PARAMETER SETTING

Parameter	Value	Remarks
N	20 ~ 50	Number of Points
l	5.0m	Length of the rope
r	0.05m	Radius of the rope
m	0.05kg	mass of one point
G	9.8N/kg	Gravity
k_h	1200	Virtual coupling spring constant
s	0.003	Scale factor for output force
μ	10	Friction constant
k_{rs}	100	Repulse spring constant
k_{rd}	5	Repulse spring damper constant
k_l	800	Linear spring constant
k_d	1	Linear damper constant
k_{ts}	10	Torsional spring constant
k_{td}	0.05	Torsional damper constant
k_{sw}	0.2	Swivel damper constant

With the parameters above, we can obtain around $500Hz \sim 1000Hz$ update rate for both Phantom Omnis. Users can feel the output forces of smooth quality.

B. Experiment of Knotting

We build five different models with various combinations of forces models described in section III. With two PHANTOM Omni haptic devices, users can tie an arbitrary knot about the rope which is hung up on one fixed frame.

1) *Model 1*: This model contains only a linear spring and a linear damper. It is the least realistic model. The two connected segments can bend to any angle effortlessly.

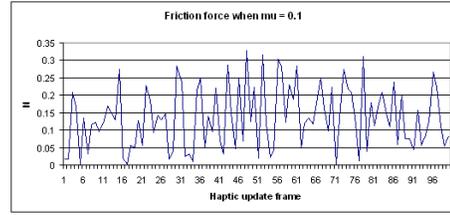


Fig. 15. Friction plot when $\mu = 0.1$

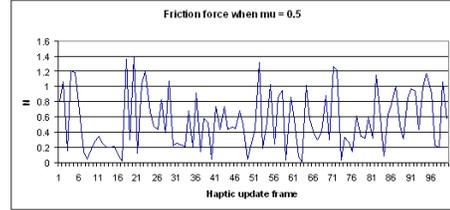


Fig. 16. Friction plot when $\mu = 0.5$

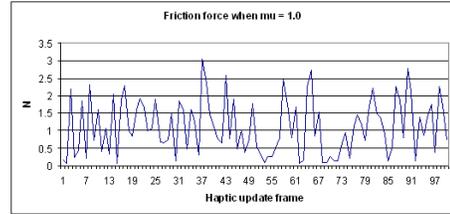


Fig. 17. Friction plot when $\mu = 1.0$

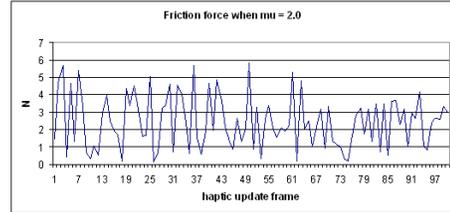


Fig. 18. Friction plot when $\mu = 2.0$



Fig. 19. Screen shot of one-hand pulling

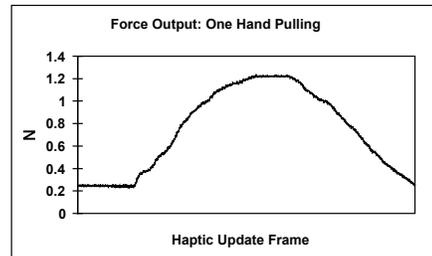


Fig. 20. Output force plot of one-hand pulling

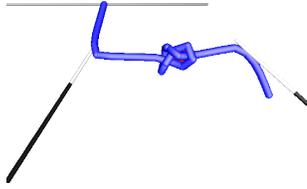


Fig. 21. Screen shot of knotting

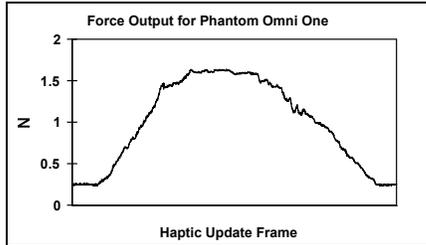


Fig. 22. Output force plot of left hand

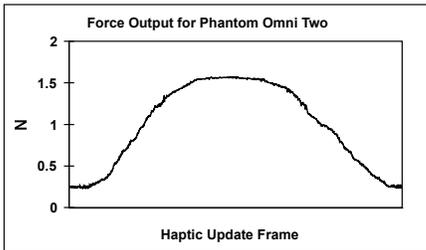


Fig. 23. Output force plot of right hand

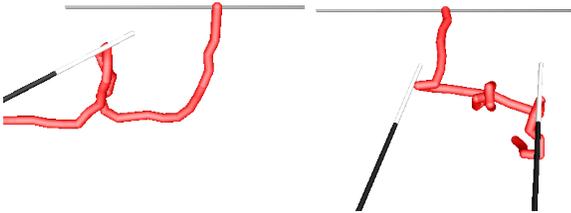


Fig. 24. Rope model 1

2) *Model 2*: This model is almost the same as model 1, but also contains a torsional spring. The torsional spring adds a lot more realistic behaviour to the thread, but also, because it uses a nonlinear function ‘acos’, it creates some harmonic wave motions.

3) *Model 3*: Compared to model 2, a torsional damper has been added to this model. This damper stops the harmonic motion presented in model 2. But this model creates another class of instability where it is very sensitive to the thread and creates a self-excitation phenomenon.

4) *Model 4*: This model includes a ‘swivel’ damper to fix the problem of perpetual orbiting (the self excitation mentioned in the above). The result is a thread that looks more like a real thread.



Fig. 25. Rope model 2

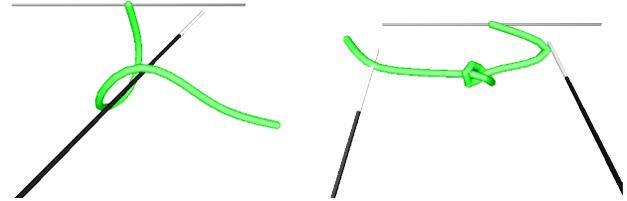


Fig. 26. Rope model 3

5) *Model 5*: This model has all the components of model 4. The only difference is that the linear spring’s force computed quadratically on the difference between its current length and rest length, instead on linearly. This makes the thread appear a lot less stretchy, which is more realistic since the real threads stretch very little. The thread’s non-linear response also makes it a lot more responsive to movements.

Comparing the results from above five different models, we can draw a conclusion that model 4 is the most ideal model for our surgical training environment.

C. Experiment of Unknotting

Same as the knotting experiment, the rope is hung up on one fixed frame. Also, to make knotting and unknotting easier, we set up a desk under the rope model to let part of the rope lay on the desk. In order to untie a knot successfully, we have to pick up the right point, otherwise the knot could be more tightening instead of loosening. This is part of the unknotting planning algorithm which will not be discussed here. Fig. 29 show the successful unknotting of a over-hand knot and a figure-of-eight knot. Fig. 30 shows if you grab the wrong point, the knot can not be untied.

VII. CONCLUSION AND FUTURE WORK

We presented a fast and simple approach to compute 3D DLO simulations. We simulate both internal forces and external forces. Also, we analyzed how forces propagate along the rope when the user pulls the rope with one hand or two hands. While our simulation cannot produce physically exact shapes and forces, even sometimes the user might feel the force feedback a little unstable (because of the high demanding of the haptic device refresh rate), our methods can be used in virtual reality simulation to give users more realistic senses.

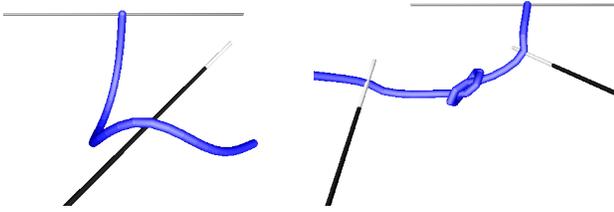


Fig. 27. Rope model 4

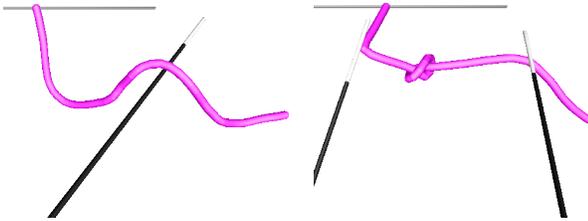


Fig. 28. Rope model 5

Because our model is based on the finite element method, to make ropes more realistic, we must add more segments and more mass points to the model, which may cause the program run more slowly (the more mass points the model has, the more time we need to complete dynamic computation and collision detection). Therefore, we cannot guarantee the haptic rendering rate to be around 1000Hz. Users may feel the force output less smooth sometimes. To solve the problem mentioned above, we may introduce level of detail methods to the modeling and undertake some optimization of dynamic computation and collision detection methods. Also, to speed-up the operation, we will look into Physics Process Unit (PPU) for the case of suturing and knotting in surgical training environment.

In this paper, we did not consider the static friction, our next step is to study the forces when the user is trying to tie a knot tightly and untie a tight knot.

Also, in the future, we will study how this rope model will interact with deformable objects such as human tissue, which is one of the most important parts of surgical simulations.

REFERENCES

[1] H. Wakamatsu, E. Arai, S. Hirai, *Knotting/Unknotting Manipulation of Deformable Linear Objects*, International Journal of Robotics Research, Volume 25, Issue 4 (April 2006), Pages: 371 - 395, ISSN:0278-3649

[2] H. Wakamatsu, K. Takahashi, S. Hirai, *Dynamic Modeling of Linear Object Deformation based on Differential Geometry Coordinates* Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on Publication Date: 18-22 April 2005 On page(s): 1028- 1033 ISBN: 0-7803-8914-X

[3] J. Takamatsu, T. Morita, K. Ogawara, H. Kimura, K. Ikeuchi, *Representation for Knot-Tying Tasks*, Robotics, IEEE Transactions, Volume: 22, Issue: 1, On page(s): 65- 78, Feb. 2006

[4] M. Saha, P. Isto, *Motion planning for robotic manipulation of deformable linear objects* Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on May 15-19, 2006 Page(s):2478 - 2484

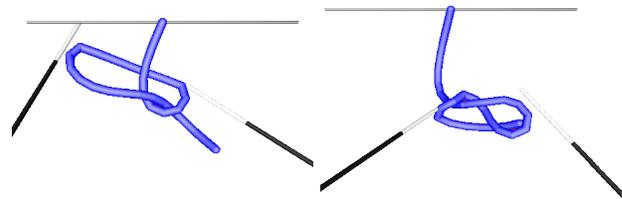


Fig. 29. Success unknotting

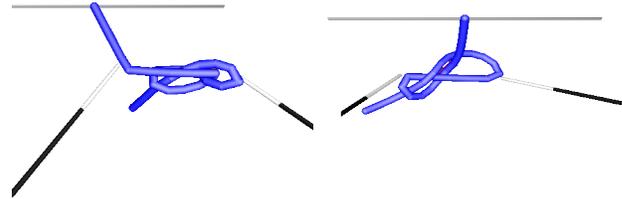


Fig. 30. Unsuccessful unknotting

[5] D. Pai, *Strands: Interactive simulation of thin solids using cosserat models*, Computer Graphics Forum, 21(3):347-352, 2002. Proceedings of Eurographics'02.

[6] C. Wang, A.M.D. Richardson, D. Liu, R. Rosing, R. Tucker and B. De Masi, *Construction of Nonlinear Dynamic MEMS Component Models Using Cosserat Theory*, Proc. SPIE Design, Test, Integration & Packaging of MEMS Symposium, 2003

[7] D. Q. Cao, Dongsheng Liu, Charles H.-T. Wang, *Three Dimensional Nonlinear Dynamics of Slender Structures: Cosserat Rod Element Approach*, eprint arXiv:math/0410286,10/2004

[8] M. Grgoire, E. Schmer, *Interactive simulation of one-dimensional flexible parts*, Proceedings of the 2006 ACM symposium on Solid and physical modeling. Pages: 95 - 103 Year of Publication: 2006 ISBN:1-59593-358-1

[9] J. Phillips, A. Ladd and L.E. Kavraki, *Simulated knot tying*, Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on Volume 1, 11-15 May 2002 Page(s):841 - 846 vol.1

[10] B. Kahl, D. Henrich, *Manipulation of deformable linear objects: Force-based simulation approach for haptic feedback*, 12th International Conference on Advanced Robotics (ICAR 2005), July 18th-20th, 2005

[11] F. Wang, E. Burdet, A. Dhanik, T. Poston, C. L. Teo, *Dynamic Thread for Real-Time Knot-Tying*, Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems - Volume 00 2005

[12] M. LeDuc, S. Payandeh, J. Dill, *Toward modeling of a suturing task*, Graphics Interface (GI), pp 273-279, Halifax, Nova Scotia, 2003.

[13] J.E. Colgate, M.C. Stanley, J.M. Brown, *Issues in the haptic display of tool use*, Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on Publication Date: 5-9 Aug 1995

[14] J. Lenoir, P. Meseure, L. Grisoni, C. Chaillou, *A rope model for surgical simulation*, 2nd International Symposium on Medical Simulation (ISMS'04) pp. 105-113, Cambridge, Massachusetts (USA).

[15] J. Lenoir, P. Meseure, L. Grisoni, C. Chaillou, *Surgical thread simulation*, Modelling & Simulation for Computer-aided Medicine and Surgery, (MS4CMS), November 2002

[16] J. Brown, J. Latombe, K. Montgomery, *Real-time knot tying simulation*, The Visual Computer: International Journal of Computer Graphics, 20(2): 165-179.

[17] R. Bridson, R. Fedkiw, J. Anderson, *Robust treatment of collisions, contact and friction for cloth animation*, Proceedings of the 29th annual conference on Computer graphics and interactive techniques. Pages: 594 - 603 Year of Publication: 2002 ISBN ISSN:0730-0301, 1-58113-521-1

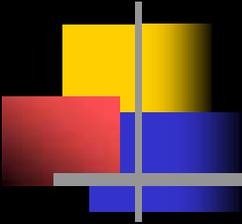
Identification of Deformation

Presentation 1: *Modeling and Identification of Rheological Deformation*

Pages: 34–90

Presentation 2: *On Compression Model for Integrative Analysis of
Different View Breast Xrays*

Pages: 91–95



Modeling and Identification of Rheological Deformation

Hiroshi Noborio
Department of Computer Science
Osaka Electro-Communication University

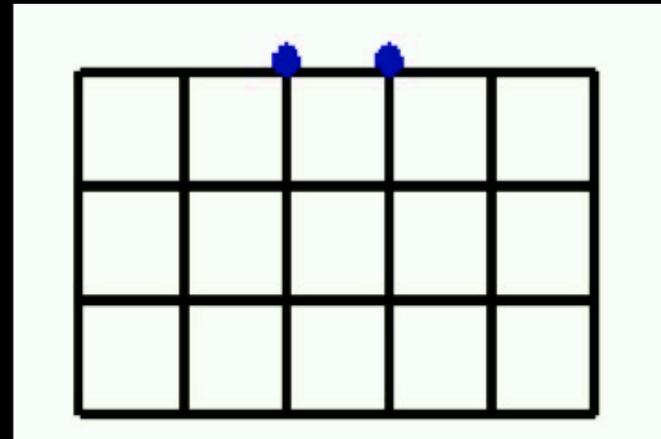
**Workshop on
Modeling, Identification, and Control of Deformable Soft Objects
to be held on November 2 (Friday)
in conjunction with
IROS 2007, Oct.29 - Nov.2, 2007, San Diego, U.S.A.**

Shape Outline

- Introduction and Research purpose
- **Model 1** – A voxel/lattice model under many basic MSD elements
- **Models 2 and 3** by adding each of local and global volume constant conditions into the **Model 1**
- Calibrating unknown parameters of each model by **RA (Randomized Algorithm)** and **GA (Genetic Algorithm)**
- Experimental comparisons
- Conclusions and future works

Introduction

- **Modeling rheology object** is a younger field.
- The model is useful to manipulate an object **by a robot arm in a real world**.
- The model is useful to feel its reactive force **by a haptic device** or to watch shape deformation **in a 3-D graphics world of PC**.



Force propagation and shape deformation should be quickly calculated. Precision and time are trade-off.

Research Purpose

- Elastic and Visco-elastic objects have been modeled by **Mass Spring Damper (MSD) Method**, **Finite Differential Method (FDM)**, **Boundary Element Method (BEM)**, **Finite Element Method (FEM)**

Concerning to FEM

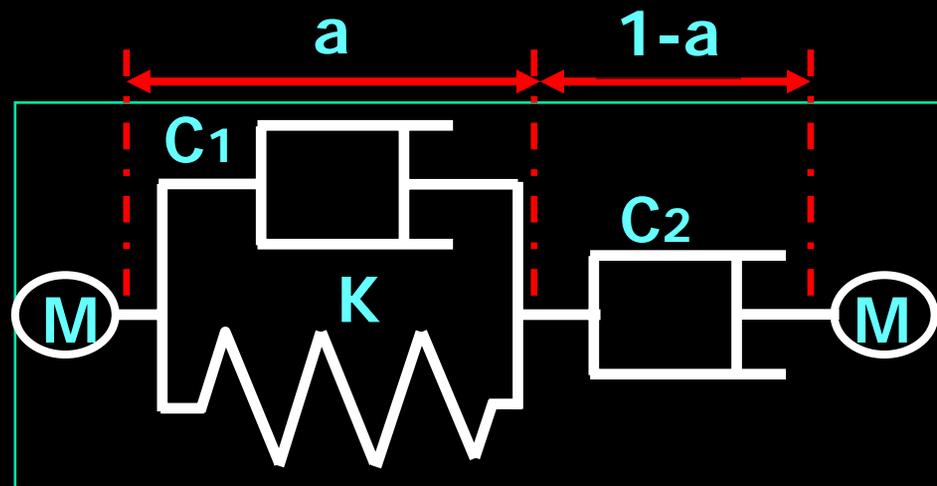
- Force propagation and shape deformation are precise.
- Calculation time is enormous.

Concerning to MSD

- Calculation time is small enough.
- Force propagation and shape deformation are not so precise.

MSD Basic Element

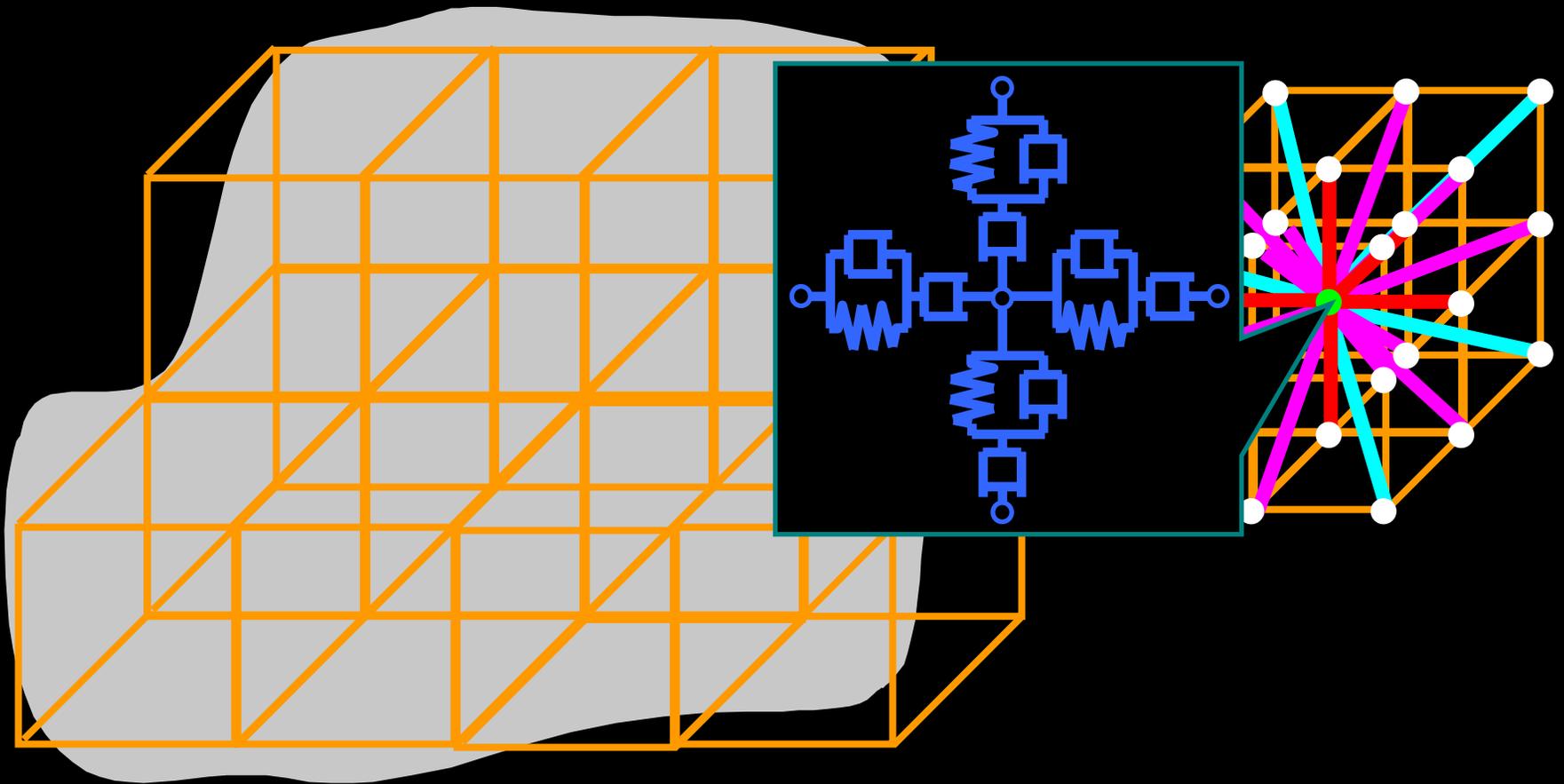
Basic element consists of
Voigt model and damper.



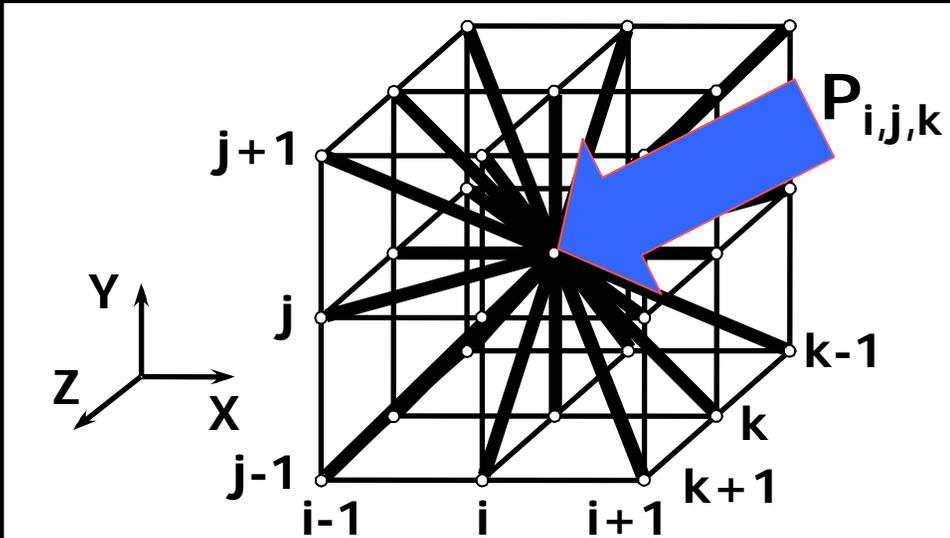
Basic element

Rheology property (e.g., residual displacement) is flexibly condensed.

Voxel/Lattice Basic Model



Constructing Dynamic Equation



Internal forces between $P_{i,j,k}$ and its neighbor masses

$$F_{i,j,k}^{int} = \sum_{(i',j',k') \in \{(-1,0,0), (0,0,0), (1,0,1)\}} F_{i',j',k'}$$

$$M = M_{object} / N$$

M_{object} : Mass of object

N : Number of mass points

$P_{i+1, j+1, k+1}$:

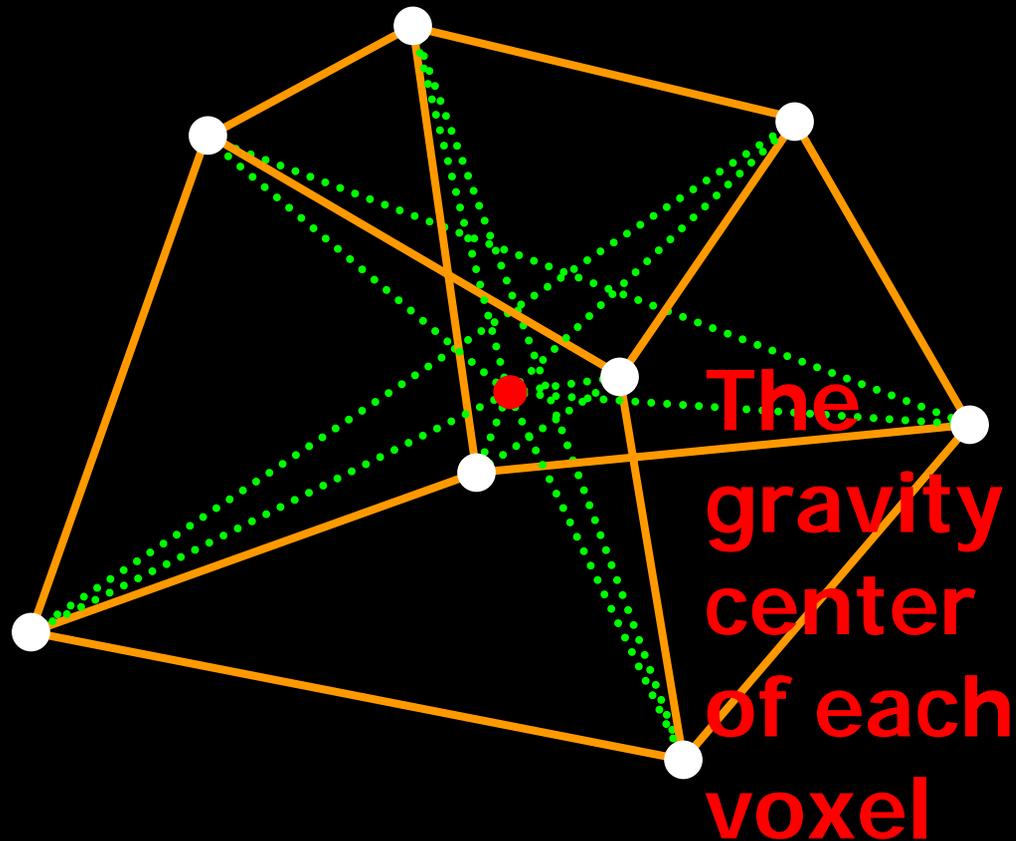
$P_{i,j,k}$ Neighbors for the mass point

$$MP_{i,j,k} = F_{i,j,k}^{int} + F_{i,j,k}^{ext}$$

External forces directly

The equation is solved by integration methods, e.g., RK, Midpoint, Euler, BDF methods.

Local Volume Constant Condition

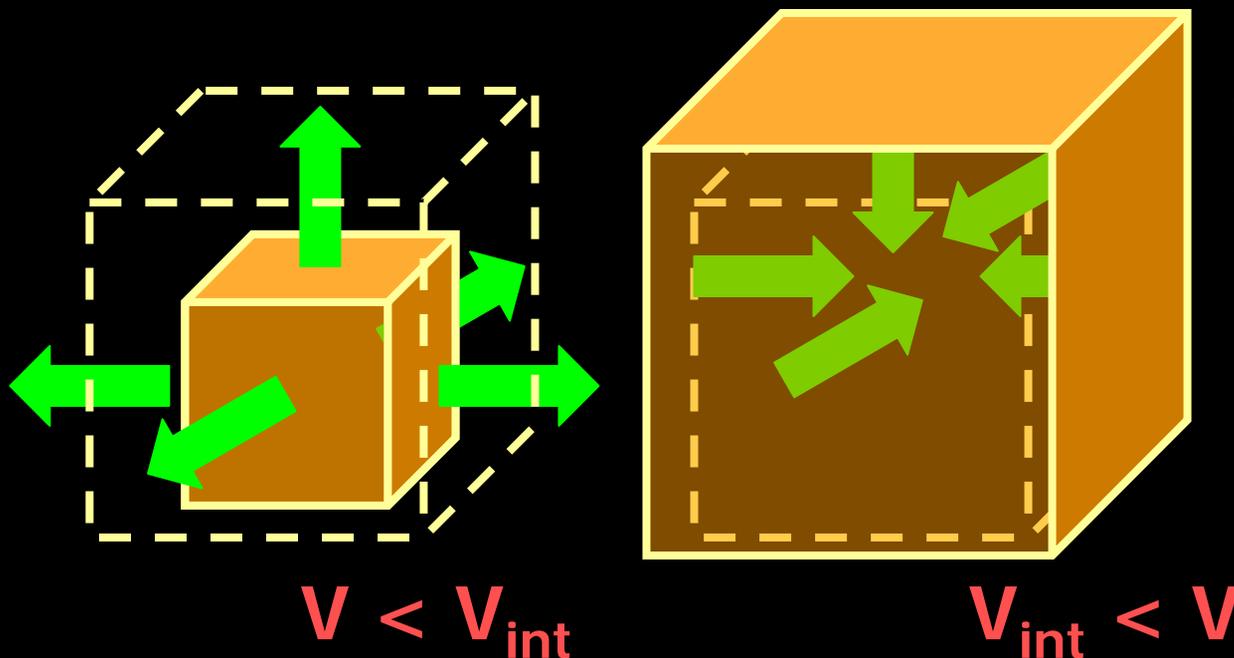


Global Volume Constant Condition

In order to converge virtual volume to real one, we always give an external force p around virtual object.

Feedback force under Pascal's principle

$$p = -k_{vol} (\dot{V} - V_{int}) - c_{vol} \dot{V}$$



Elastic coefficient: k_{vol}

Viscous coefficient: c_{vol}

Present volume: V

Initial volume: V_{int}

Difference of \dot{V} : \dot{V}

Measuring Shape Deformation of Real Rheology Object

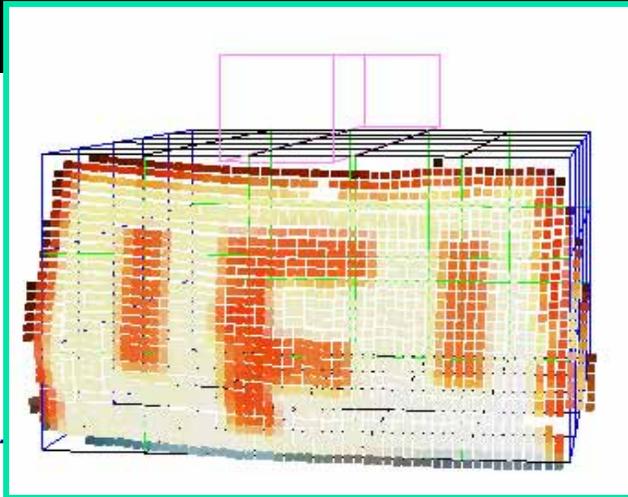


- **Real Rheology Object**
- A real rheology object is made by mixing wheat flour and water.
- The volume of rheology object is $10 \times 6 \times 10 = 600[\text{cm}^3]$.

■ Real-time stereo visions

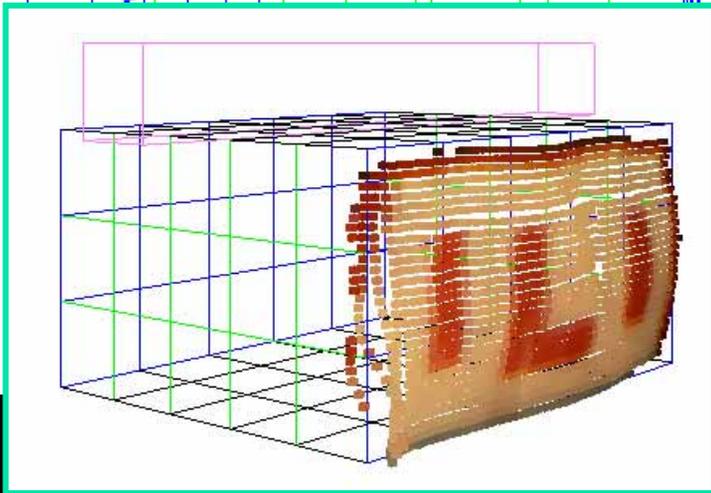
- A set of surface points whose number is about 1000 is captured in real-time manner.

Virtual Rheology Object

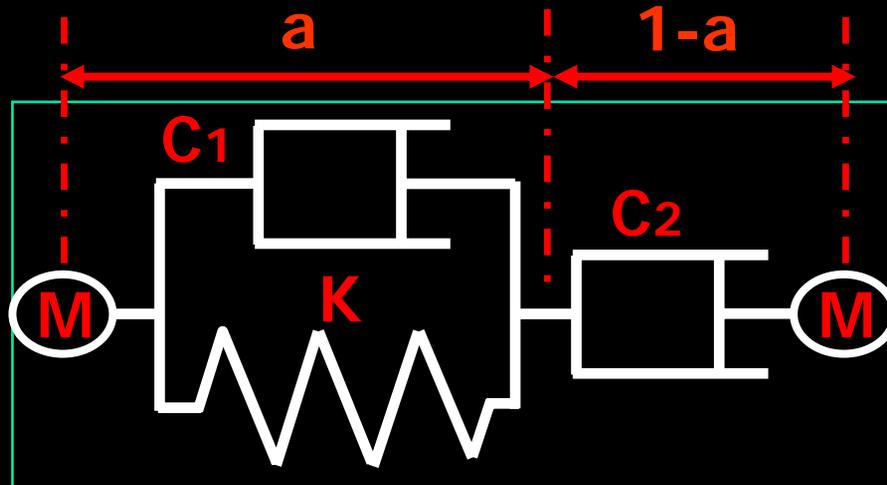


- How to measure the difference between virtual and real objects

- A set of captured points are initially located on a virtual object.
- The minimum distance from a captured point to a virtual object is calculated by a modified LCA.
- The sum S of distances from captured points at four times are calculated.



Parameters to Calibrate



Our MSD basic element

- Elastic coefficient K
- Viscous coefficient C_1, C_2
- Length ratio between Voigt and the other parts a

In order to minimize the sum S of distance errors at four times, we calibrate uncertain parameters.



Randomized Algorithm (RA)
Genetic Algorithm (GA)

Randomized Algorithm (RA)

1. Initialize all the uncertain parameters $K, C1, C2$ and a within their intervals.
2. We select a local minimum by the **steepest descendent method**.
3. If the past time amounts to a threshold T_{cal} ($=168$ [hour]), this finishes.
4. Otherwise, after $(=10)$ is randomly added into one of many parameters T_{ran} ($=100$ [number]) times, return to step 2.

Genetic Algorithm (GA)

1. Initialize G_{ind} , G_{gen} , G_{eli} and G_{mut} .
2. G_{ind} is the number of individuals. Each consists of calibrating parameters. P_{gen} and G_{gen} are present and threshold generations.
3. If P_{gen} amounts to G_{gen} , GA finishes.
4. We calculate shape differences S_n ($n=1,2,\dots, G_{ind}$) between real and virtual objects.

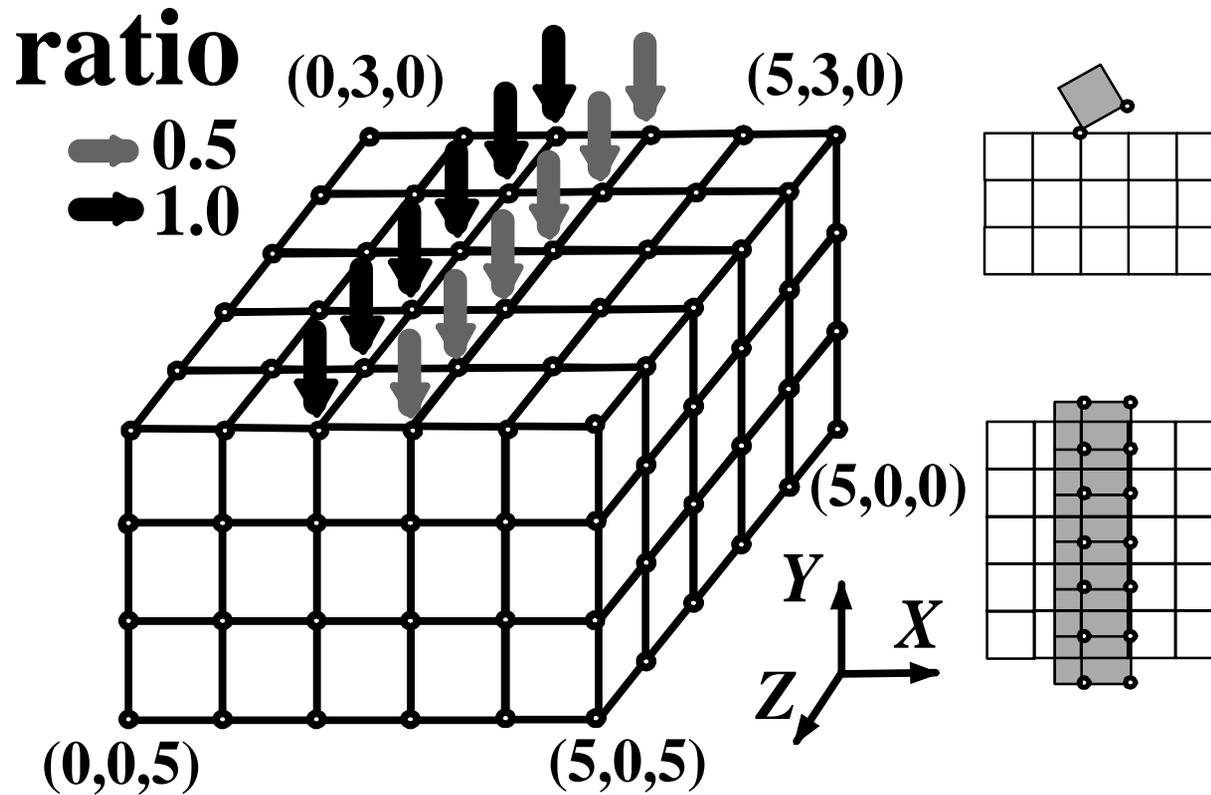
Genetic Algorithm (GA)

- [Selection]** After sorting individuals by S_n , we select better individuals whose number is $G_{ind} \times G_{eli}$ ($0.6 < G_{eli} < 1.0$).
- [Mutation]** We generate individuals by reversing bits of their originals. whose number is $G_{ind} \times G_{mut}$ ($0.0 < G_{mut} < 0.05$).
- [Crossing]** We cut and combine parts of two individuals to make the other ones.
- After increasing P_{gen} by 1, we return to 2.

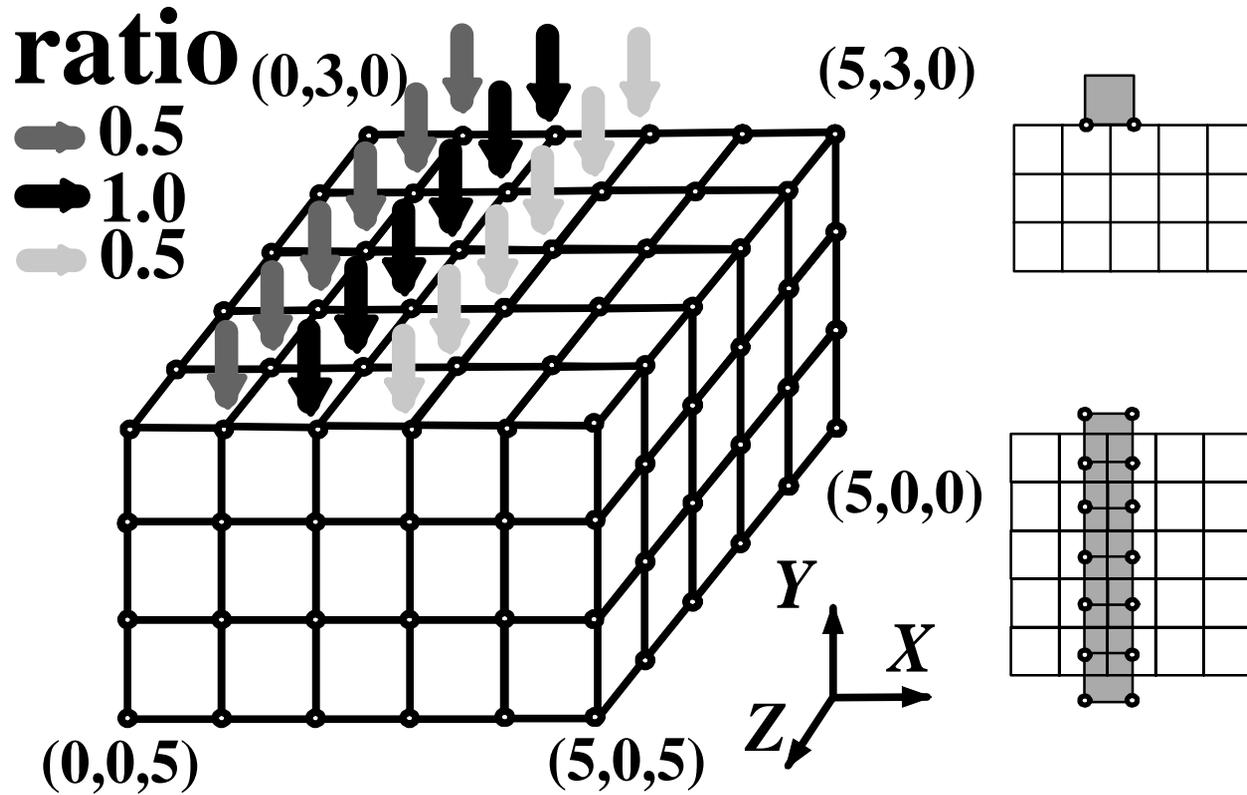
PC Circumstance

- 3-D graphics acceleration board
GeForce FX 5600, 128MB
- PC (CPU : Pentium4 3.00GHz,
Memory : 2048MB)
- 3-D graphics software Open-GL

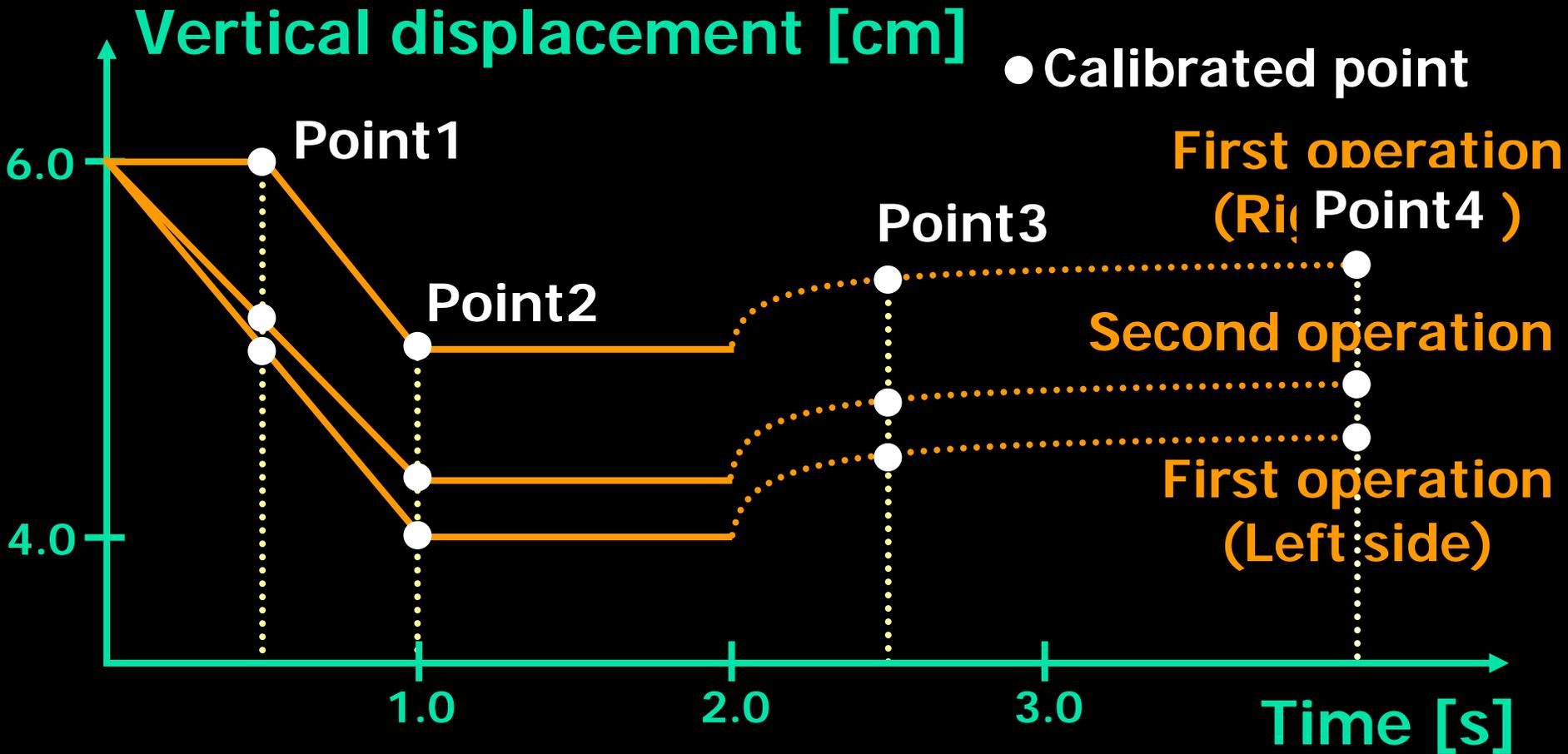
First Operation



Second Operation

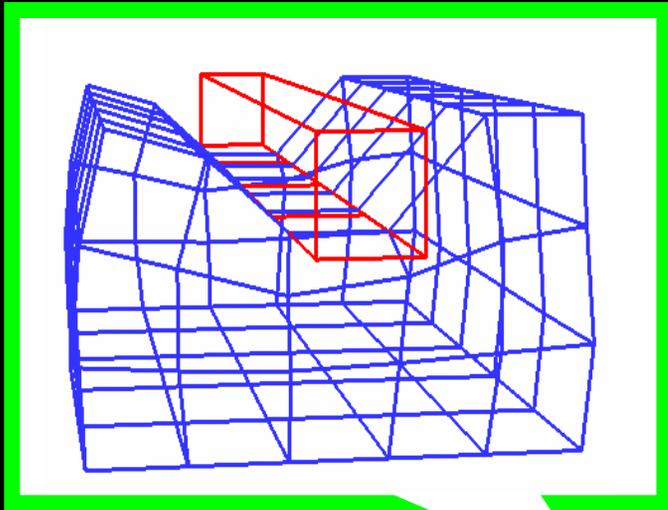


Time Schedule of Two Operation



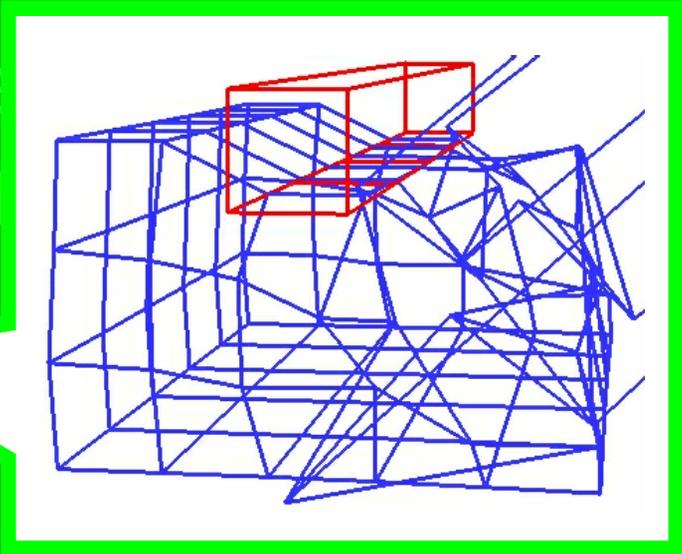
Stable Parameter Intervals

Otherwise, deformation shapes are almost destroyed.



msec

r all mode



100	K	3000[g
500	C ₁	100000
500	C ₂	200000
0.3	a	0.7

Search Space

RA $= 10$
 $T_{ran} = 100$ [number]

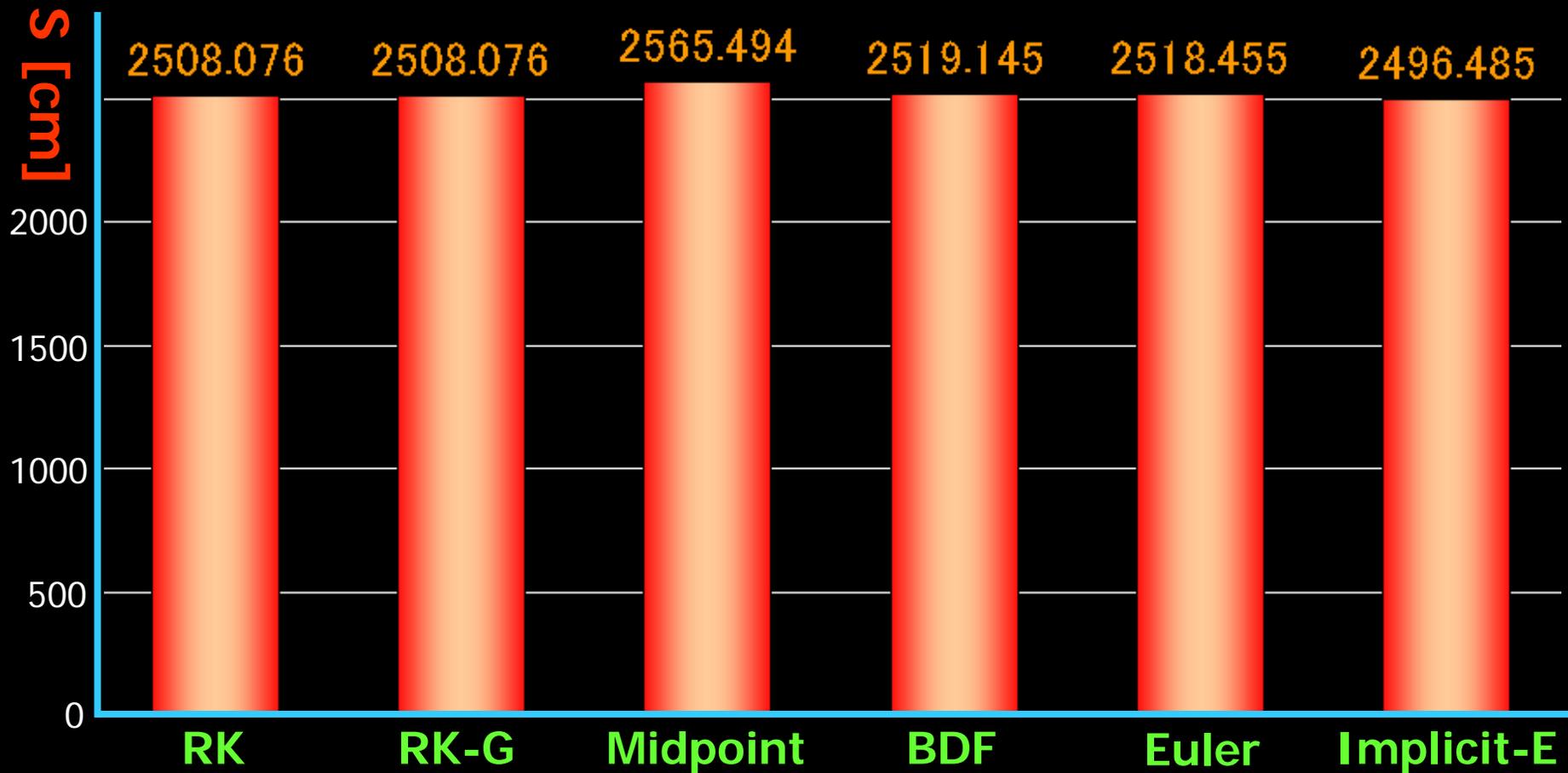
GA $G_{ind} = 50$
 $G_{eli} = 0.2$
 $G_{mut} = 0.01$

Calculation
time is
always the
same.

To keep comparative fairness of **RA** and **GA**,
we synchronously select G_{gen} and T_{cal} .

Model1: $G_{gen} = 1900$ ($T_{cal} = 168$ [hour])

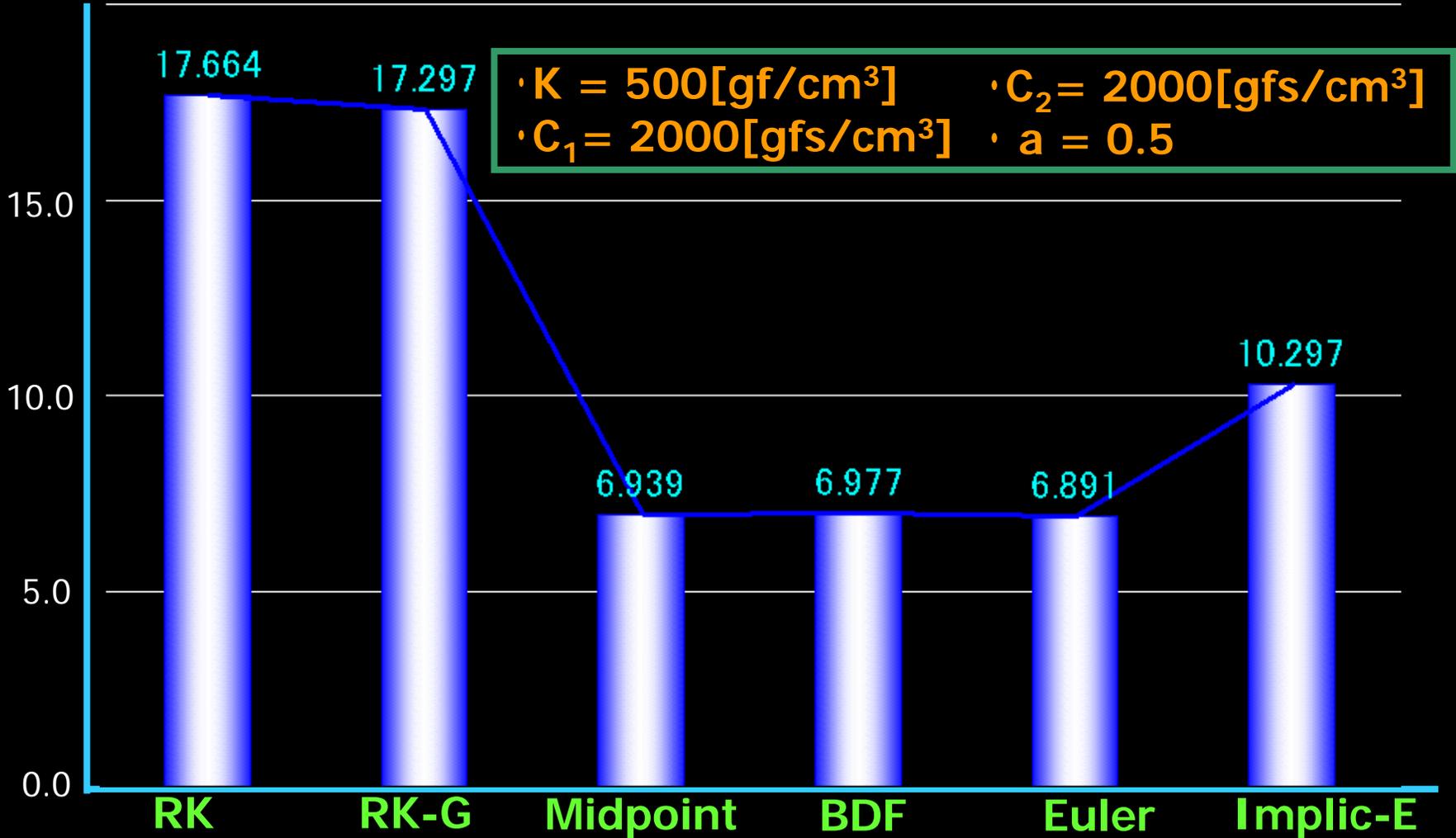
Shape Difference



· $K = 500[\text{gf}/\text{cm}^3]$ · $C_2 = 2000[\text{gfs}/\text{cm}^3]$
· $C_1 = 2000[\text{gfs}/\text{cm}^3]$ · $a = 0.5$

Integration Time

Calculation time [sec]



Four kinds of MSD models

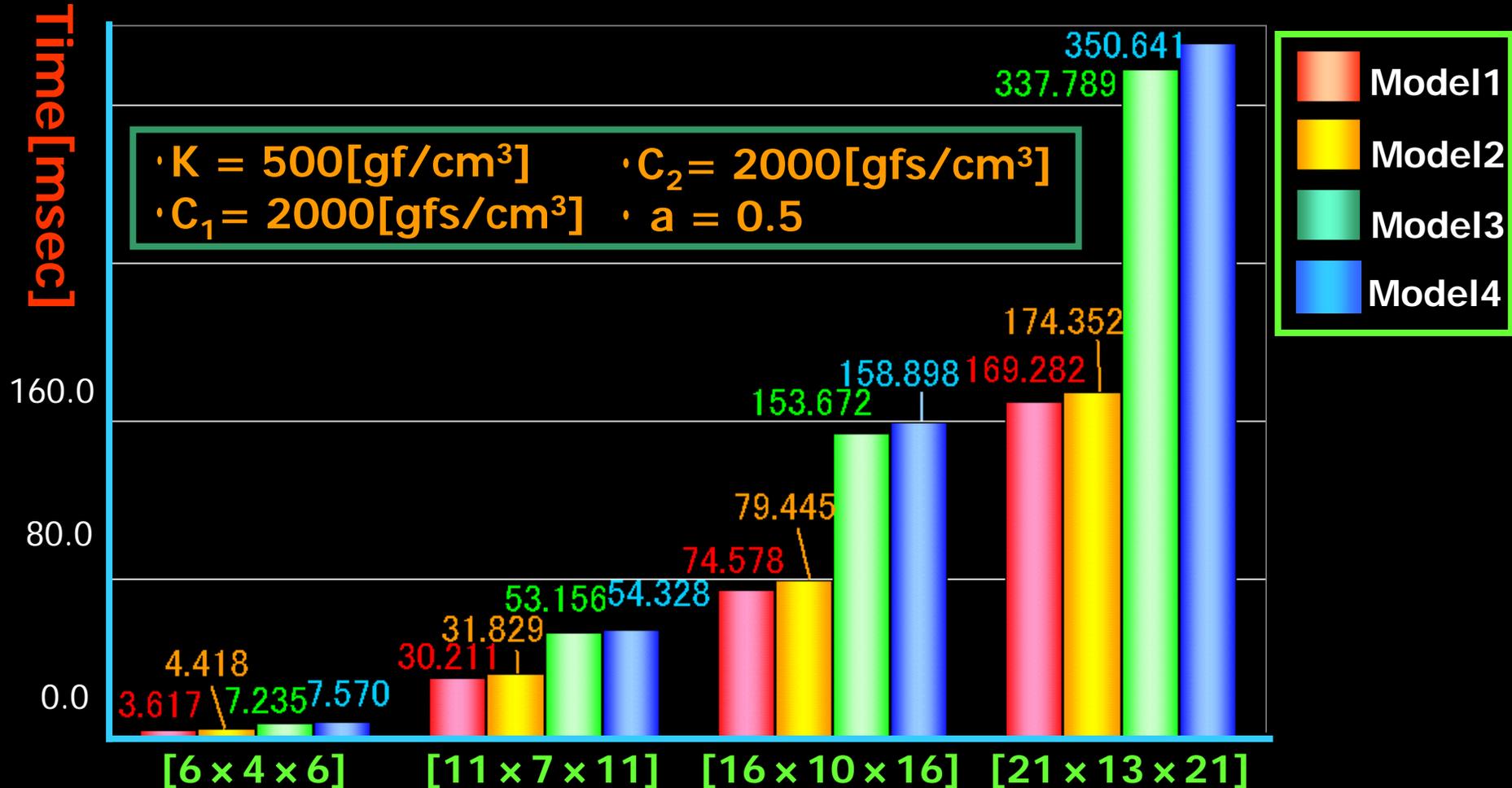
- Model1 : without volume constant condition
 - Model2 : with local volume condition
 - Model3 : with global volume condition
 - Model4 : with both volume conditions
-
- Fast calculation because the number of basic elements is slightly larger
 - Hard calibration of shortest elements in each voxel

Four kinds of MSD models

- Model1 : without volume constant condition
- Model2 : with local volume condition
- Model3 : with global volume condition
- Model4 : with both volume conditions

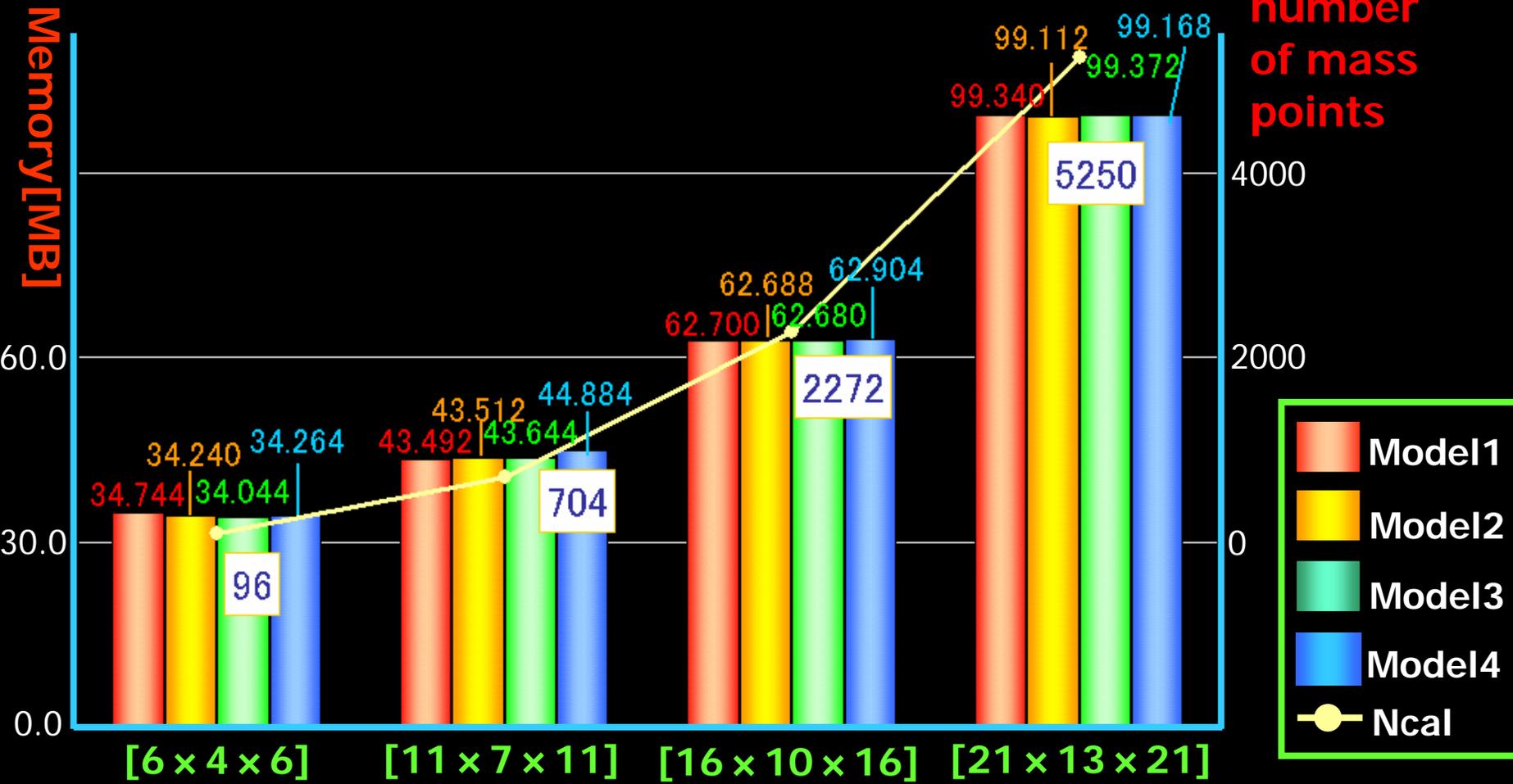
- Few calibration because of feedback property
- Hard calculation since volume should be always calculated

Calculation Time



Memory Storage

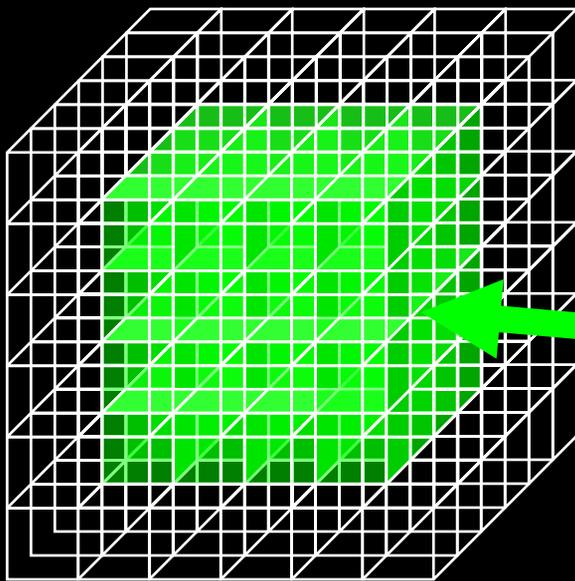
The number of mass points



$\cdot K = 500[\text{gf}/\text{cm}^3]$ $\cdot C_2 = 2000[\text{gfs}/\text{cm}^3]$
 $\cdot C_1 = 2000[\text{gfs}/\text{cm}^3]$ $\cdot a = 0.5$

Surface and Core Areas

Different force propagations in **surface** and **core** areas are assumed. Surface parameters K^{surf} , C_1^{surf} , C_2^{surf} , a^{surf} and core ones K^{core} , C_1^{core} , C_2^{core} , a^{core} should be calibrated.



A set of voxels facing object surface is defined as **surface** area.

A set of the other voxels is defined as **core** area.

The number of masses: $6 \times 6 \times 6$

Shape Comparison between RA and GA in the Model 1

The number of captured points is $N = 14551$

4 Parameters

	S	Volume	K	C_1	C_2	a
RA	2481	67.99	2256	648	15944	0.59
GA	2463	67.87	3000	3691	20000	0.70

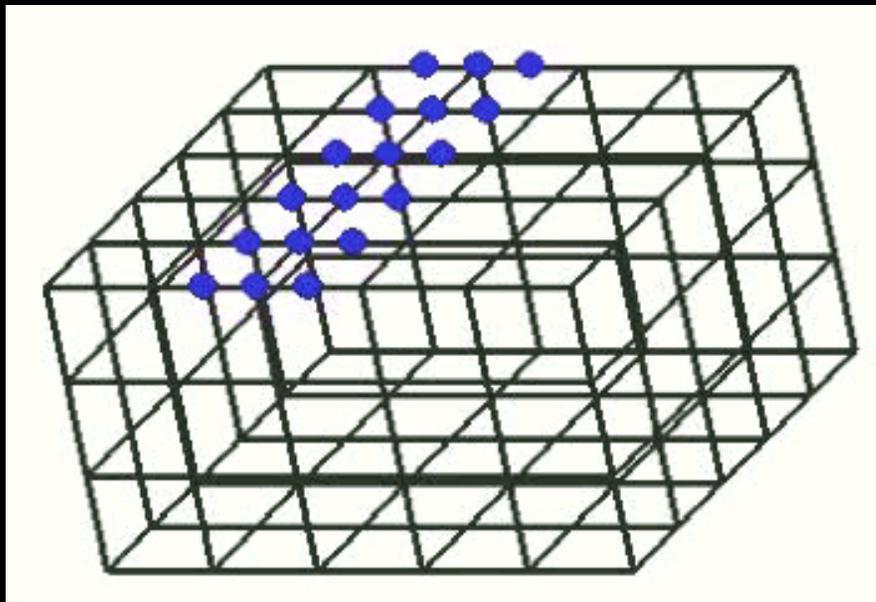
8 Parameters

	S	Volume	K^{core}	C_1^{core}	C_2^{core}	a^{core}
RA	2442	67.35	2245	4970	4277	0.68
GA	2350	67.85	293	9988	19961	0.63

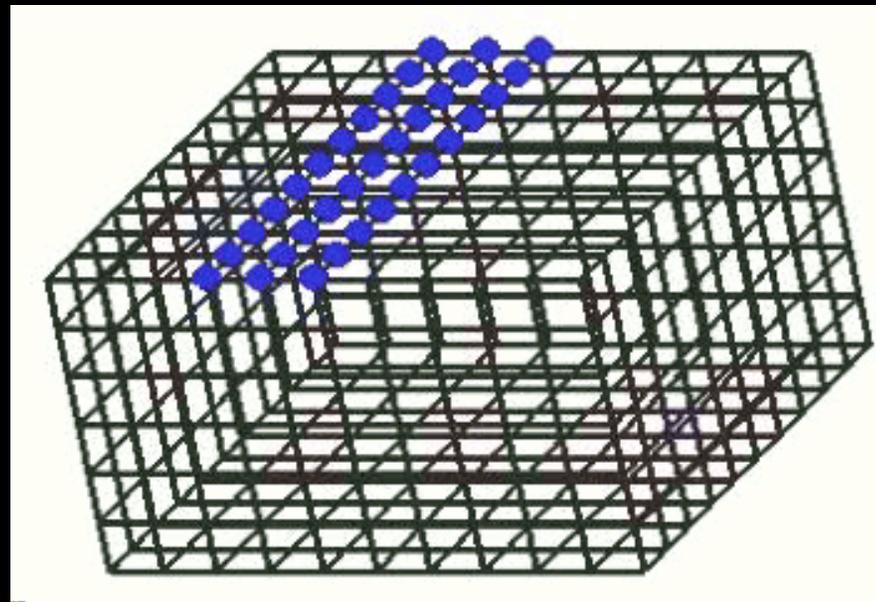
	K^{surf}	C_1^{surf}	C_2^{surf}	a^{surf}
RA	1870	3026	2526	0.47
GA	2477	561	4454	0.63

External Force Transfers in sparse and dense models

Virtual Rheology Object



Mass number: $6 \times 4 \times 6$
Lattice length: $1.0[\text{cm}]$
: Pushed masses



Mass number: $11 \times 7 \times 11$
Lattice length: $0.5[\text{cm}]$
: Pushed masses

Number of individuals in GA when model calculation is changed

G_{gen} is selected while keeping the same calibration time $T_{cal}=168$ [hour]. Thus, comparison fairness against RA is maintained.

- Model 2 as Sparse Model : $G_{gen}=1900$
- Models 3, 4 as Sparse Model : $G_{gen}=1100$
- Model 2 as Dense Model : $G_{gen}=270$
- Models 3, 4 in Dense Model : $G_{gen}=140$

If the volume is always calculated, calculation increases and generation decrease in calibration.

Number of individuals in GA when model calculation is changed

G_{gen} is selected while keeping the same calibration time $T_{cal}=168$ [hour]. Thus, comparison fairness against RA is maintained.

- Model 2 as Sparse Model : $G_{gen}=1900$
- Models 3, 4 as Sparse Model : $G_{gen}=1100$
- Model 2 as Dense Model : $G_{gen}=270$
- Models 3, 4 in Dense Model : $G_{gen}=140$

If resolution increases, calculation increases and generation decrease in calibration.

Calibrate Parameters in Local Volume Constant Condition (Models 2 and 4)

50 $K^{\text{surf-in}}$, $K^{\text{core-in}}$,
 3000[gf/cm^3]

100 $K^{\text{surf-on}}$, $K^{\text{core-on}}$,
 3000[gf/cm^3]

250 $C_1^{\text{surf-in}}$, $C_1^{\text{core-in}}$
 10000[gfs/cm^3]

500 $C_1^{\text{surf-on}}$, $C_1^{\text{core-on}}$
 10000[gfs/cm^3]

250 $C_2^{\text{surf-in}}$, $C_2^{\text{core-in}}$
 20000[gfs/cm^3]

500 $C_2^{\text{surf-on}}$, $C_2^{\text{core-on}}$
 20000[gfs/cm^3]

0.3 a^{surf} , a^{core} , a^{surf} , a^{core} 0.7

Sixteen parameters
 should be calibrated.

Parameters in each voxel

$K^{\text{surf-in}}$, $C_1^{\text{surf-in}}$,
 $C_2^{\text{surf-in}}$, $a^{\text{surf-in}}$,
 $K^{\text{core-in}}$, $C_1^{\text{core-in}}$,
 $C_2^{\text{core-in}}$, $a^{\text{core-in}}$

Parameters on each voxel

$K^{\text{surf-on}}$, $C_1^{\text{surf-on}}$,
 $C_2^{\text{surf-on}}$, $a^{\text{surf-on}}$,
 $K^{\text{core-on}}$, $C_1^{\text{core-on}}$,
 $C_2^{\text{core-on}}$, $a^{\text{core-on}}$

Comparative Shape Results

The number of captured point is N = 15732

	S (Sparse)	S (Dense)	Volume (Sparse)	Volume (Dense)
Model 2'	1831	1506	76.30	72.88
Model 3	1724	1663	74.97	75.16
Model 4'	1835	1352	75.94	75.64

4' The number of points whose errors are larger than 0.25cm

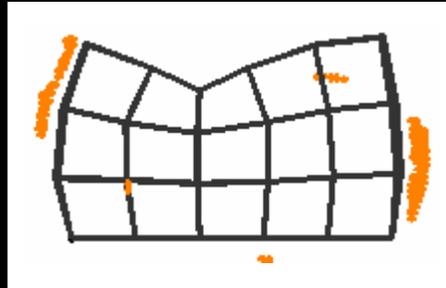
		First	Second	Third	Fourth	Total
Model 2'	Sparse	187	395	415	393	1390
	Dense	219	167	232	267	885
Model 3	Sparse	361	265	299	267	1192
	Dense	523	263	199	212	1197
Model 4'	Sparse	215	331	480	567	1390
	Dense	258	188	168	147	761

Comparative Shape Results (cont'd)

Real Rheology
Object



Model 2'



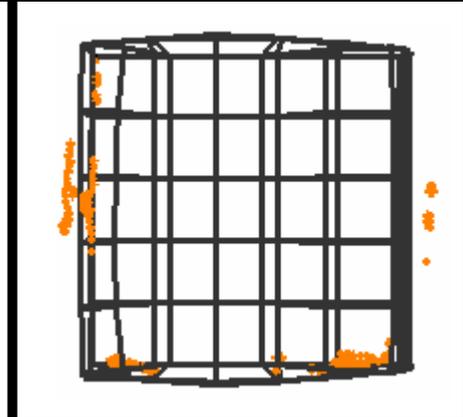
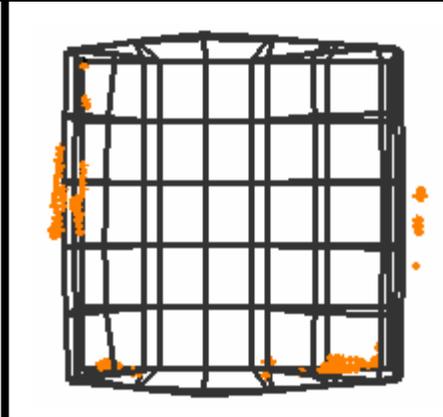
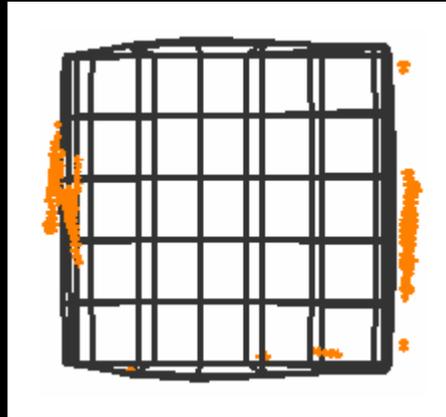
Model 3



Model 4'



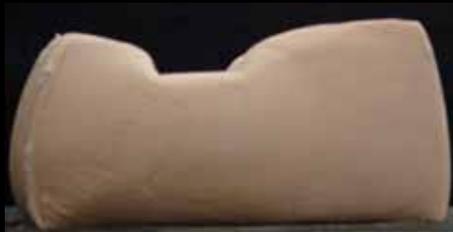
Front side



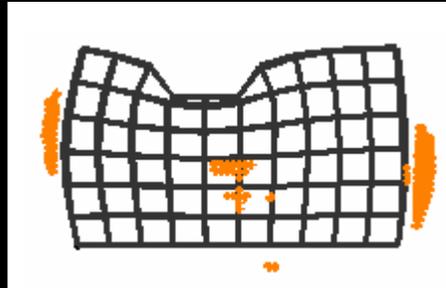
Upper side

Comparative Shape Results (cont'd)

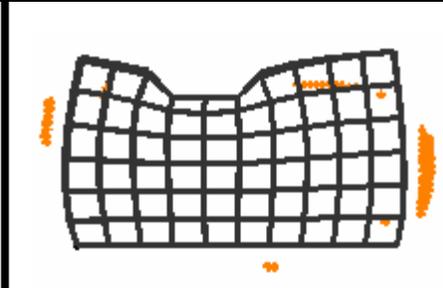
Real Rheology
Object



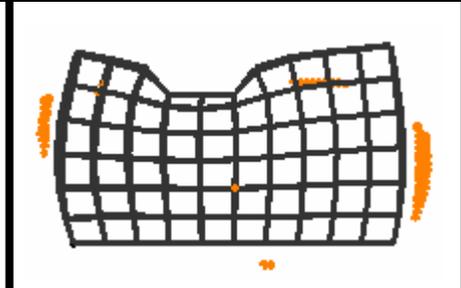
Model 2'



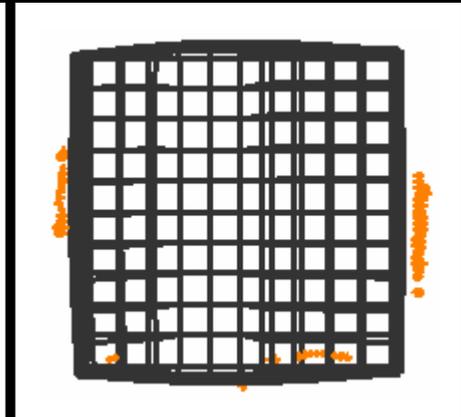
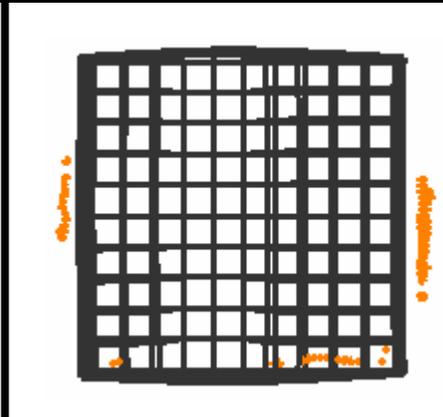
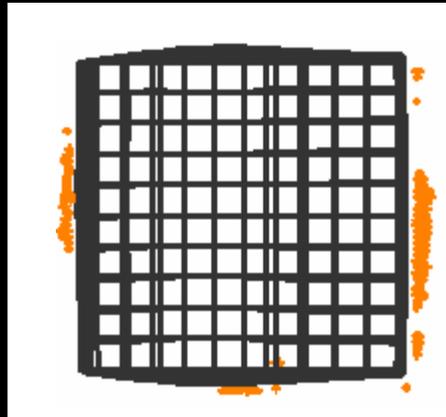
Model 3



Model 4'



Front side



Upper side

Conclusions

- By using MSD voxel/lattice models and two volume constant conditions, we build a virtual rheology object.
- Euler Method whose time step is 2 ms is selected concerning to computational efficiency and shape stability.
- If the number of uncertain parameters is larger, **GA** is better than **RA**.
- The larger the numbers of calibrating parameters and discrete voxels are, the better precision of deformation shape is.

Ongoing Works

- **For higher speed**
 - Using **octree** as hierarchical voxel model. The smaller the number of basic elements is, the smaller calculation time of force propagation is.
 - Using **FPGA** as hardware programming. A lot of same procedures in basic elements are parallel processing.
- **For better precision**
 - Using another structure such as **tetrahedral structure**. Also using another basic element such as **Voigt and Maxwell** parts. Also using **non-linear spring/damper coefficients**.

Force Outline

- Introduction & Research purpose
- **Model1** - Cell/lattice structure with many basic **MSD** (**M**ass-**S**pring-**D**amper) elements
- **Model2 & 3** by adding each of **local** & **global volume constant conditions** into **Model1**
- Classic force models
such as **Pull-off** & **Friction forces**
- **Calibrating** uncertain parameters of each model by **shape deformation** & **force impulses**
- Experimental Comparisons
- Conclusion & future works

Introduction

- Modeling of a rheology object is younger field.
- Modeling is useful
 - ... for manipulating deformable materials
 - ... for feeling contact force with support of haptic device
 - ... for watching shape deformation in a 3D graphics world

Deformable soft materials

Food



Fabric



Biomedical tissue



➡ Rheological properties

Precise force propagation & shape deformation should be quickly calculated.

Research Purpose

- Comparison of Several Types of MSD model with/without conservation laws of volume
 - ... **by giving** displacement at surrounding masses
 - ... **by calibrating & evaluating Shape difference** between real & virtual objects

Shape Precision

The model **including conservation laws of volume** is **effective** for obtaining **precise shape deformation**.

Research Purpose

- Comparison of Several Types of MSD model with/without conservation laws of volume
... **by giving** displacement at surrounding masses
... **by calibrating** & **evaluating Shape difference** between real & virtual objects

Force Precision

All calibrations **based on Shape difference** are **not enough** for feeling **reactive force** from deformed virtual rheology object.

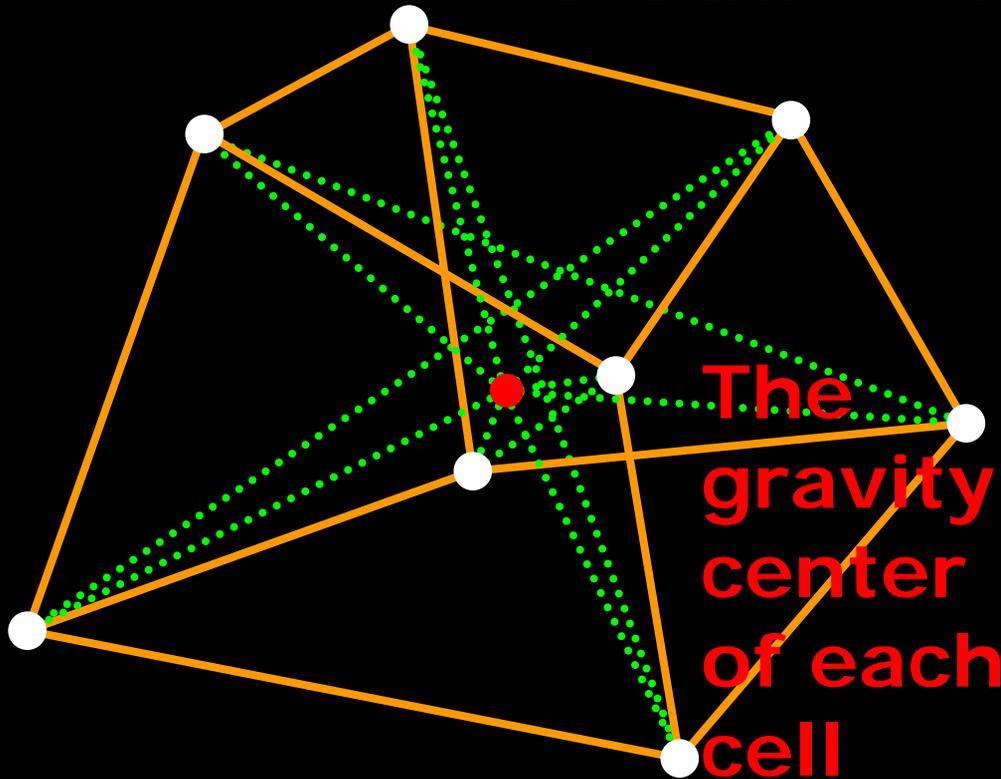
Based on Force impulses

We applied a **multiple combination** of **pushing**, **calibrating** & **evaluating** operations to MSD models

➡ **to investigate the best one** & **its properties**

Local Volume Constant Condition

Model2 – Model1 **with** Local volume constant condition



To expand each cell :

- Longer four elements are eliminated.
- Their half elements are added.



This feed-forward approach **converges** a virtual volume to a real one indirectly

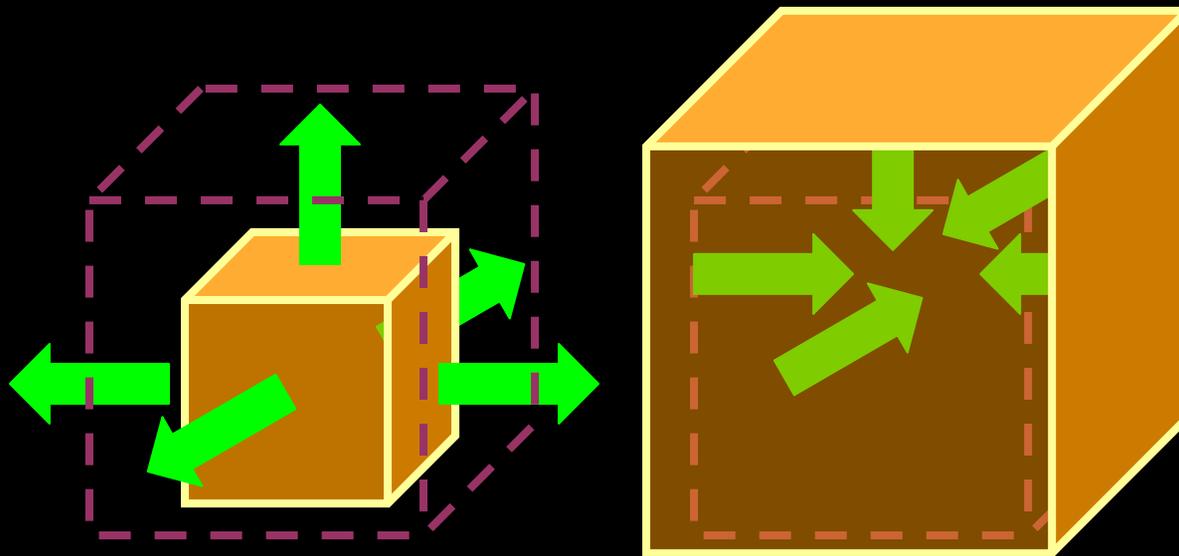
Global Volume Constant Condition

Model3 – Model1 **with** Global volume constant condition

In order to converge virtual volume to real one, we always give an external force **p** around virtual object.

Feedback force under Pascal's principle

$$p = -k_{vol} (V - V_{int}) - c_{vol} \dot{V}$$



Coefficient of
Elasticity : k_{vol}
Viscosity : c_{vol}
Present volume : V
Initial volume : V_{int}
Differential of V : \dot{V}

$$V < V_{int}$$

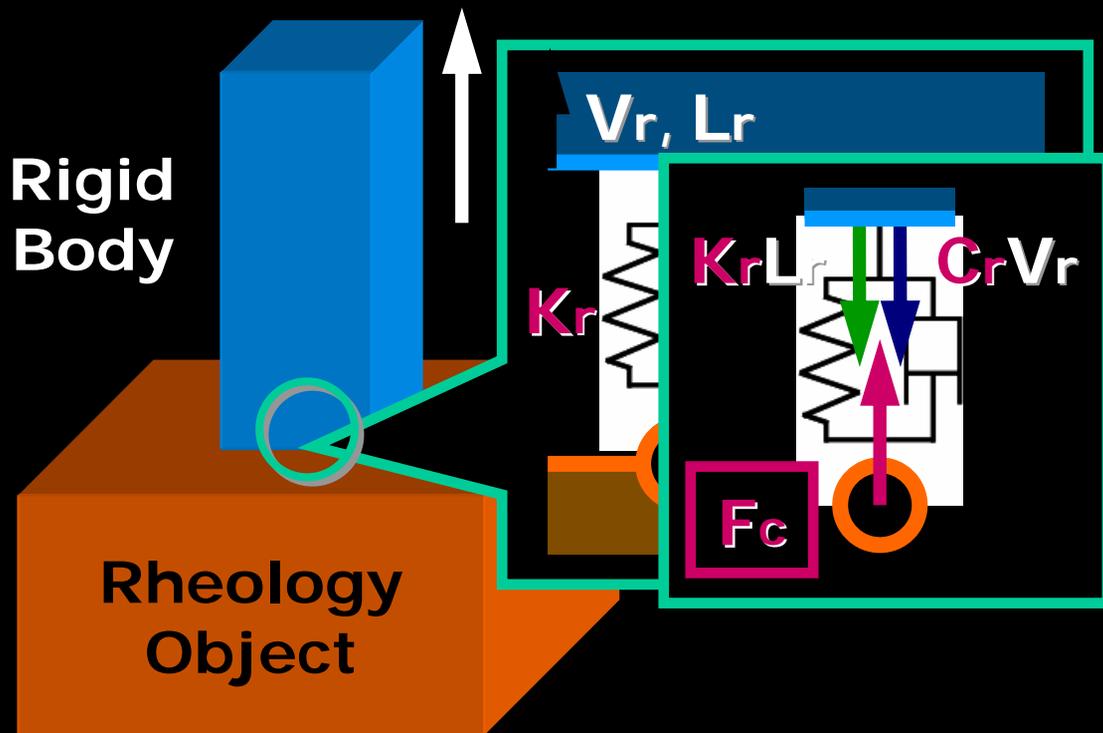
$$V > V_{int}$$

Pull-off Force

- Pull-off force : F_c [N]

If a rigid is **slowly** left from the rheology object, we should consider a pull-off force as follows :

Just Released



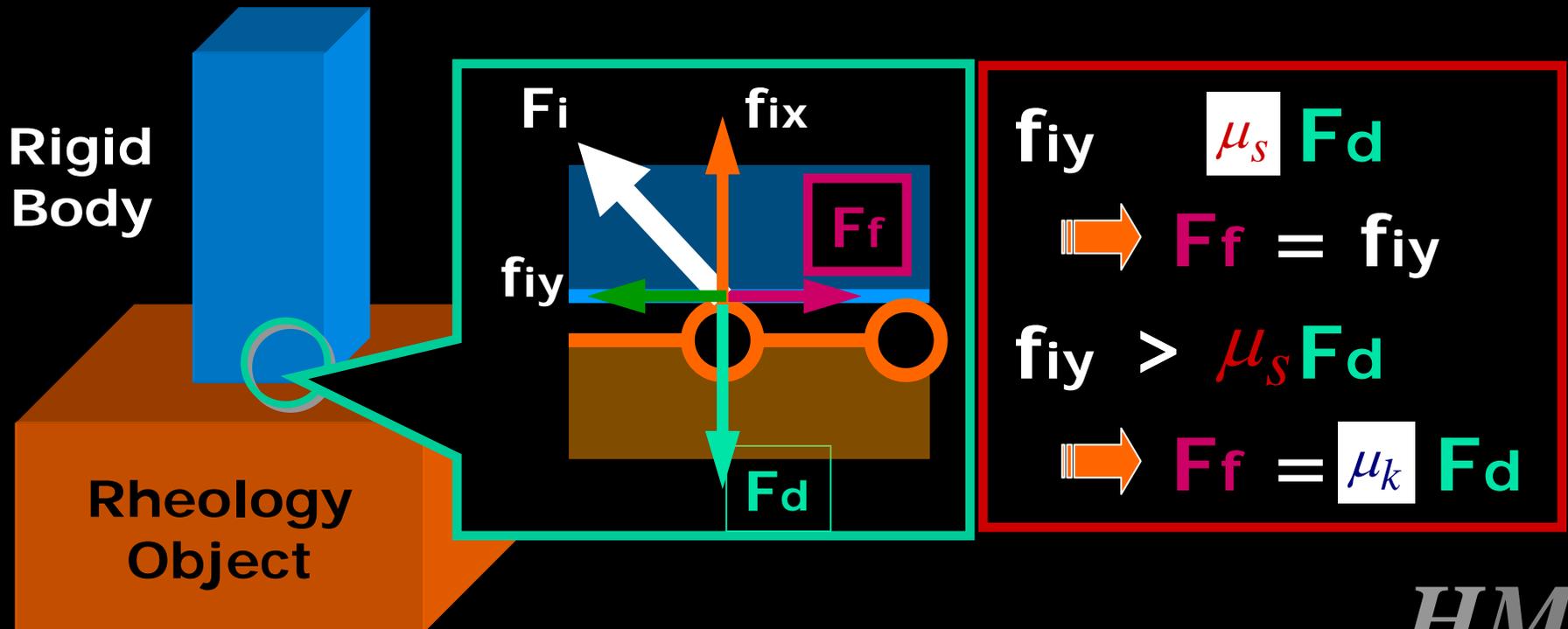
Voigt model is located at each mass point between rigid and rheology object.

Its force is neglected if the relative velocity is over a given threshold (V_t).

Friction Force

- Friction force : F_f [N]

If two object are encountered, static ($\mu_s F_d$) or kinetic ($\mu_k F_d$) friction appears between rigid body and rheology object.



Shape Measuring System



- **Real Rheology Object**
 - A real object is made by mixing wheat flour and water.
 - The volume is $10 \times 6 \times 10 = 600[\text{cm}^3]$.

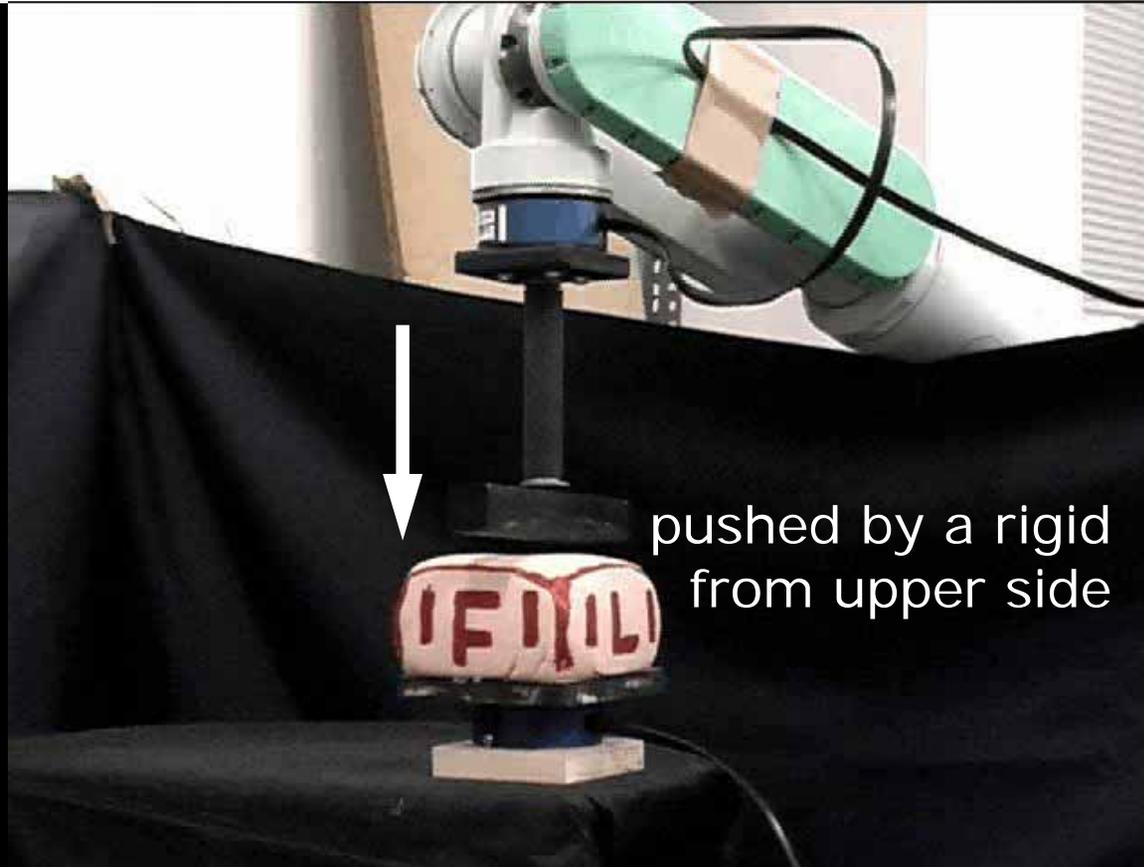
- **Real-time stereo visions** : A set of surface points whose number is about 1000 is captured at least ten times per second.
- **Difference calculation** : The sum of errors is quickly calculated by Lin-Canny algorithm.

Force Measuring System



- **Real Rheology Object**
 - A real object is made by mixing wheat flour and water.
 - The volume is $10 \times 6 \times 10 = 600 [\text{cm}^3]$.
- **Real-time force sensors** : 3 DOF forces and 3 DOF moments are captured by the sampling is 8 [kHz].

Video

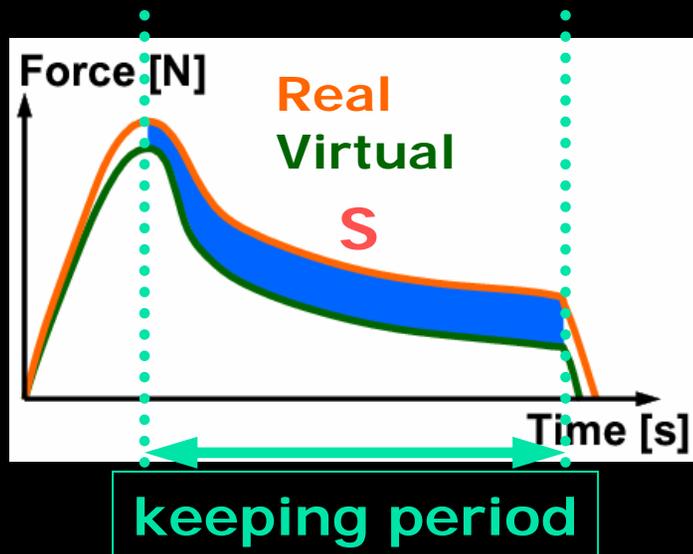


Pushing : 0.0 – 1.0 [sec]
Keeping : 1.0 – 3.0 [sec]
Releasing : 3.0 – [sec]

Parameters to calibrate

In order to calibrate many uncertain parameters of MSD models, we define **S** which is the sum of {force or shape} differences during the keeping period.

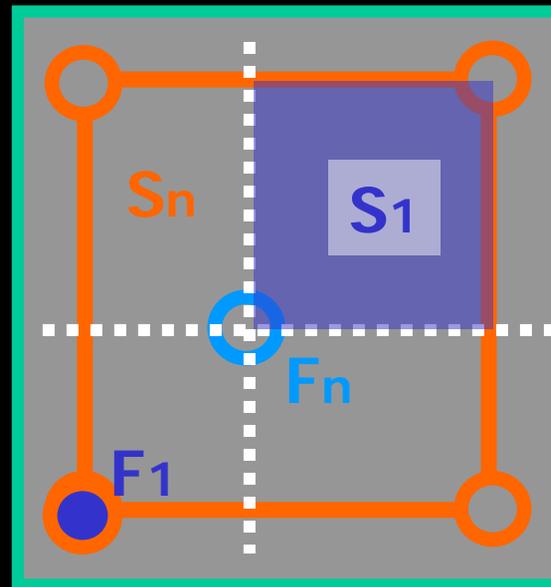
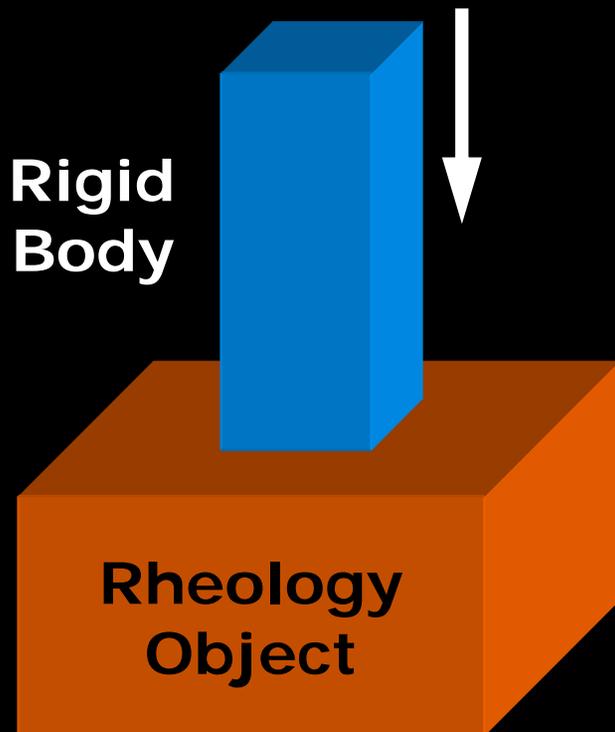
(e.g., Force differences)



By using **GA**, we determine a set of uncertain parameters so that **S** can be **minimized**.

Digital Force Transmission

- Forces or displacements are exchanged between mass points around encountered rigid and rheology objects.
- The magnitude of force or displacement is proportional to opposite areas.



Upper View

$$F_1 = F_n (S_1 / S_n)$$

Human Machine Interface Lab

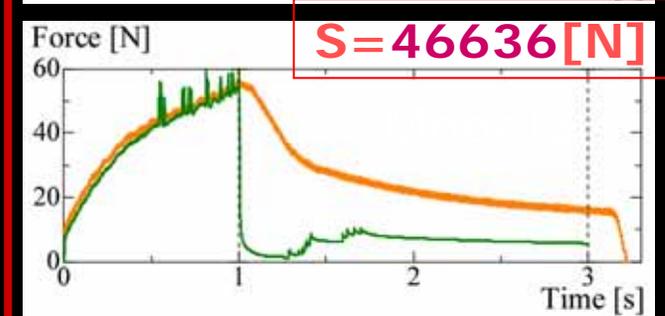
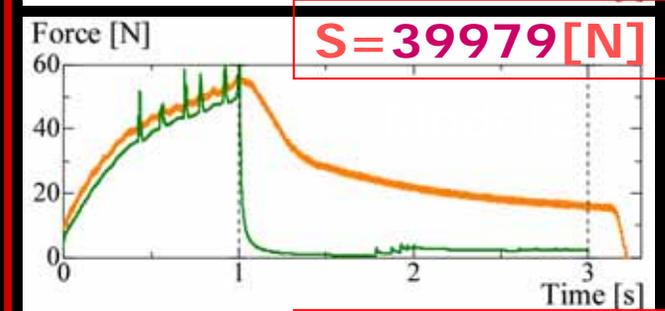
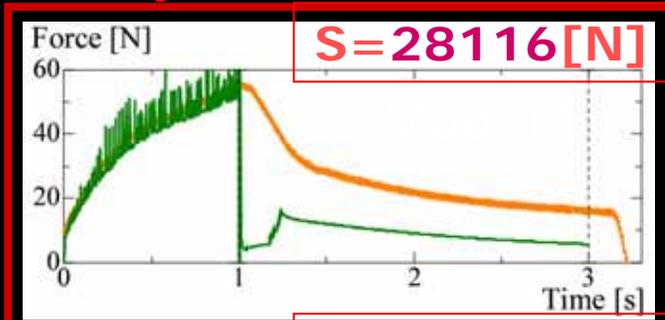
Comparisons

Three models applied by two kinds of combinations (TA, TB) are evaluated by their force differences in order to clarify their characteristics.

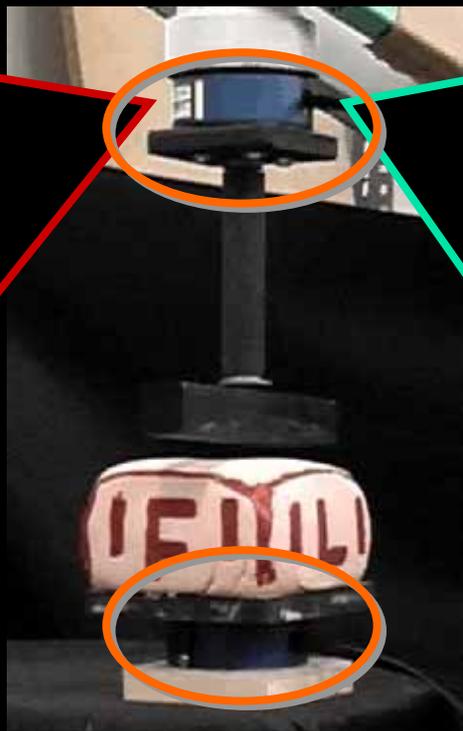
- **Three kinds of models**
 - **Model1** – Our basic model with cell/lattice structure
 - **Model2** – **Model1 with** a Local volume constant condition
 - **Model3** – **Model1 with** a Global volume constant condition
- **Two kinds of combinations**
 - **TYPE-(A)** : Pushing by **displacement** – Calibrating by **shape** – Evaluating by **force**
 - **TYPE-(B)** : Pushing by **force** – Calibrating by **force** – Evaluating by **force**

Experimental results

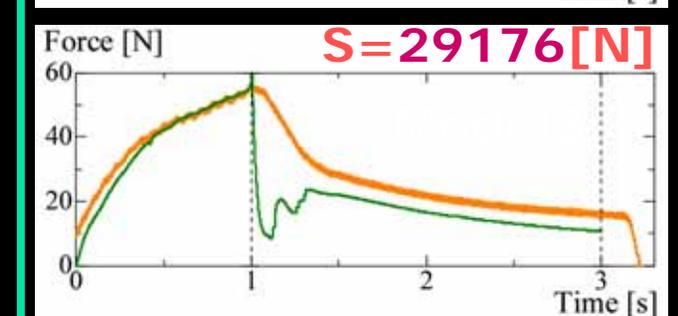
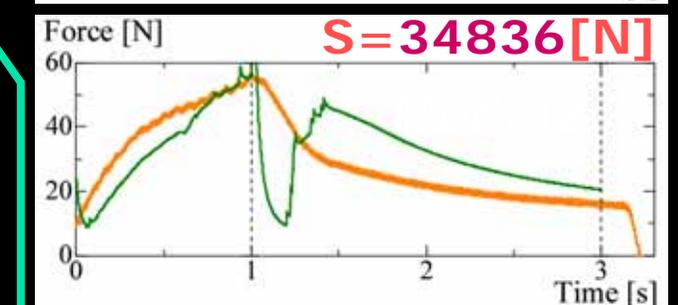
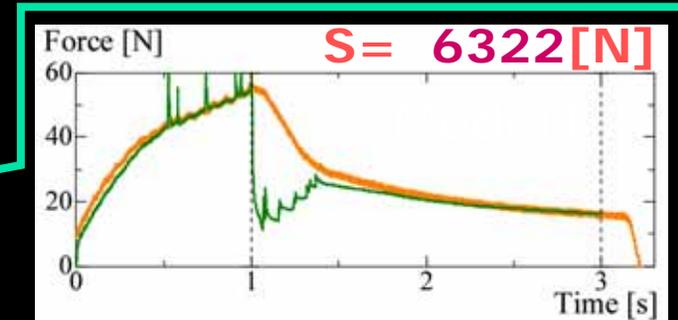
■ Evaluated by sum of **Force differences S** :
Calibrated by sum of **Shape difference (A)**



Real
Virtual



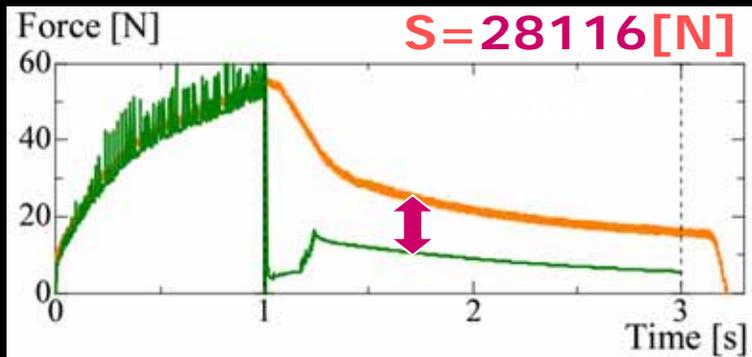
Calibrated by sum of **Force difference (B)**



Experimental results

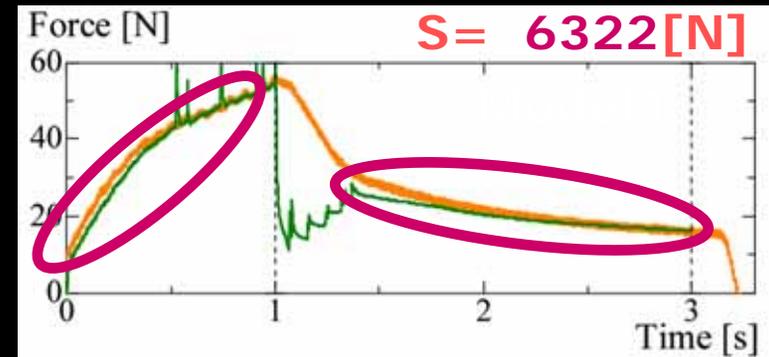
- Evaluated by sum of **Force differences S** :

Calibrated by sum of **Shape difference(A)**



Real
Virtual

Calibrated by sum of **Force difference(B)**



Model1 made by the different two combinations

Model1 made by the combination of pushing & calibrating by force sequences and their difference **shows** high precision of force.

Conclusions

- A modified **MSD** model is proposed for simulating **rheological characteristic**, and also classic pull-off, and friction models are additionally used.
- The model **excluding** volume constant condition (**Model1**) is **more suitable** for obtaining **force precision** because **no ad-hoc force** appears.
- The model made by the combination of pushing and calibrating by force sequences and their difference shows **high precision of force**.

Future Works

- To improve force precision,
 - We increase calibrated parameters.
 - We select efficient calibration algorithm.
 - We analyze force propagation directly.
- To try another set of complex pushing,
 - We use another basic element (e.g., MSD element, Voigt & Maxwell element).
 - We consider a new combination of pushing/calibration/rheology objects.

Tactile Feeling

➔ For development of
Virtual Environment



Visual Feeling

References

- Haruyuki Yoshida, Yasuyuki Murata and Hiroshi Noborio, "A Smart Rheologic MSD Model Pushed/Calibrated/Evaluated by Experimental Impulses, " Proc. of the IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp.1614-1621, 2005.
- Ryo Nogami, Fumiaki Ujibe, Hiroki Fujii and Hiroshi Noborio, "Precise Deformation of Rheologic Object under MSD Models with Many Voxels and Calibrating Parameters," Proc. of the IEEE Int. Conf. on Robotics and Automation, pp.1919-1926, 2004.
- Ryo Nogami, Hiroshi Noborio, Seiji Tomokuni and Shinichi Hirai, "A Comparative Study of Rheology MSD Models whose Structures are Lattice and Truss," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3809-3816, 2004.
- Hiroshi Noborio, Ryo Enoki, Shohei Nishimoto and Takumi Tanemura, "On the Calibration of Deformation Model of Rheology Object by a Modified Randomized Algorithm, " Proceeding of the IEEE International Conference on Robotics and Automation, pp.3729-3736, Taipei, September 15-18, 2003.
- Hiroshi Noborio, Ryo Nogami, Ryo Enoki, "Precise Deformation of Rheology MSD Model Calibrated by Randomized Algorithm," Eurographics 2003, Short Presentations, Granada Spain, pp.171-178, 2003.

On Compression Model for Integrative Analysis of Different View Breast Xrays

Yasuyo Kita, Shinichi Tokumoto, and Shinichi Hirai

Abstract—The breast deformation during medical procedures makes it difficult to analyze different breast images integratively. The simulation of the breast deformation is effective to compensate the difference of the breast shape among the images. However, realistic simulation is very difficult since the actual deformation is quite complicated and the detailed conditions about the deformation are hardly known mainly because of the large individual variations in both geometry and tissue organizations. For a CAD system for integratively analyzing different view mammograms (breast Xrays) [1], a simplified model to simulate the breast compression, which is derived base on several approximations about the deformation is used. Although it contributes to derive valuable results, the precision of the results is desired to improve. In this paper, we discussed about a breast compression model aiming at more accurate solutions. Two trials for better understanding on the breast deformation under the compression are shown: simulation of mammographic compression using a mechanical model and inspection of internal deformation using a devised phantom. These experimental results gave us some clues for better simulation.

I. INTRODUCTION

Breast cancer is one of the most serious disease for women. Recently, several types of medical images are used for screening programs to detect breast cancer in its early stages with less oversight. Especially, mammograms, MRI and Ultrasound are well used because of their good balance in trade-off between specificity/sensitivity and cost. It is effective to integrate the information obtained from different images, as each modality has its merits and demerits. However, it is not easy because a breast deformed into very different shapes during the procedures. For mammograms, a breast is strongly compressed into flattened shape primarily to reduce x-ray dosage. For taking MRI, the breast is pendulous in specially designed breast coil as the patient lies on her front in the magnet; the gravity pulled down the breast so as to get it away from her chest wall. On the other hand, since women lie on the bed with her face up during the ultrasound process, the breast shape is pulled toward the chest wall by the gravity.

It is a natural and sensible strategy to simulate the deformation of the breast to compensate the difference in the shapes among the images. At first, deformable models which

deform so as that image features of the two different images coincide each other were used to compensate relatively small deformation, such as pre-post contrast MRI[2][3] [4]. Later, to decrease deformations inconsistent to the physics, constraints based on volume conservation principle were introduced[5][6].

For larger deformation as like seen in compression procedures, finite element models (FEM) of breast have been intensively studied recently aiming at accurate simulation[7][8][9][10][11][12][13]. However, FEM models can simulate accurate deformation only when all internal and external factors, such as geometry and biomechanics of the breast, forces exerted to the breast, and boundary conditions are given correctly. Unfortunately, in most of practical cases, it is quite difficult to grasp these information: geometry of the breast varies largely according to individuals; organization of internal soft tissues are also different among the individuals; both strength of compression and boundary condition at the connection of chest wall are very complicated.

Especially, in the case of integrative analysis of different view mammograms (breast Xrays), the input is only two projective images of differently deformed breast. To simulate the breast compression process for building a CAD (Computer Aided Diagnosis) system under such situation, Kita et al [1] introduced a simplified compression model based on several approximations. For appropriate simplification, it is important to understand the principle and tendency of the breast deformation under compression. Actually, the compression model was derived based on several clinical studies[14],[15]. Especially, Novak[15] studied deformations of the breast surface during compression by observing the movements of marks made on the skin of volunteers' breast. However, such observations only give information about skin movement, and do not strictly give information about movements of tissue internal to the breast. To improve the model, we need more investigations about the internal deformation caused by the compression.

In this paper, we showed some preliminary trials for better understanding of the breast deformation including its internal deformation to improve the compression model used in the CAD system. In Section II, we briefly explain the scheme of the CAD system and the compression model used in the current system. In Section III, the breast deformation by compression is investigated by comparing a mammogram and a MRI of the same breast through the intermediary of a mechanical model. In Section IV, the internal deformation by compression is investigated by observing a phantom in a industrial CT scanner, which is devised to have 286 points of

Y. Kita is with Information Technology Research Institute, National Institute of Advanced Industrial Science and Technology, 1-1-1 Umezono, Tsukuba 305-8568, Japan y.kita@aist.go.jp

S. Tokumoto is with System Division, Industrial Technology Center of Wakayama Prefecture, Ogura 60, Wakayama city, Wakayama pref., 649-6261, Japan tokumoto@wakayama-kg.go.jp

S. Hirai is with Dept. Robotics, Ritsumeikan Univ. Kusatsu, Shiga 525-8577, Japan hirai@se.ritsumeai.ac.jp

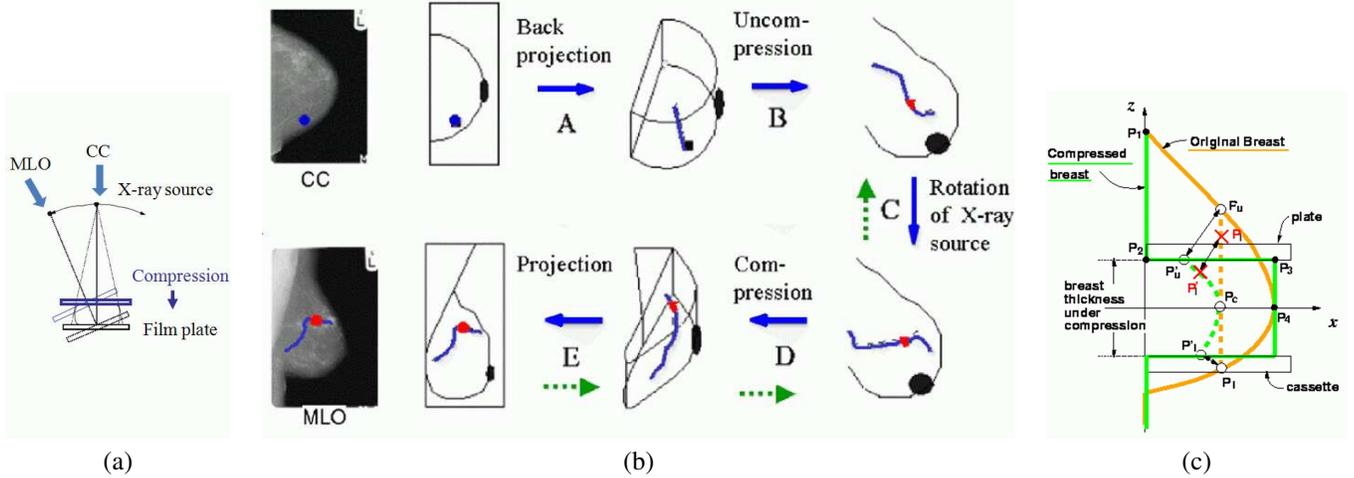


Fig. 1. Strategy for determining 3D position from two mammographic views.

marks inside of it. Finally, we discuss and clarify the matters to improve in Section V.

II. SCHEME OF THE CAD SYSTEM [1]

Recently, performing two different views of the breast, the medio-lateral oblique (MLO) and crano-caudal (CC), is spread in screening programs, since it may greatly improve sensitivity and specificity. When a mammogram is performed the breast is compressed between the film cassette and compression plate in the direction of the x-ray source: “head to toe” for the CC view and “over the shoulder diagonally to the hip” for the MLO view, as shown in Fig. 1(a). Unfortunately, radiologists find it difficult to relate points in the CC view to those in the MLO view because the breast is largely deformed in the different directions. To help their diagnosis, a CAD (Computer Aided Diagnosis) system which simulates the breast deformation during mammogram performance and suggests the corresponding position between different viewed mammograms has been developed[1]. The method calculates the epipolar curve, that is the locus of possible corresponding positions of the point in the other image by simulating the five steps of the process: A: back projection \rightarrow B: uncompression \rightarrow C: rotation \rightarrow D: compression \rightarrow E: projection as shown by the solid arrows as shown in Fig. 1(b). As a result, the line in the MLO image is calculated.

This physics-based approach have another merit: it also can help to estimate the 3D position of a lesion in the uncompressed breast, despite the fact that the breast is never imaged in the uncompressed state in mammography. This technique works after finding the corresponding point along the epipolar curve and then back-tracking the movement of the point during the simulation of the processes as shown by the dashed arrows in Fig. 1(b). Fig. 2 shows an example of the resultant 3D position obtained by the system. Here, the 3D shape of the individual breast shown in the right-hand two windows is automatically reconstructed from the breast outlines in CC and MLO images. This localization is very important to guide biopsies and/or fusion of multi-modal data

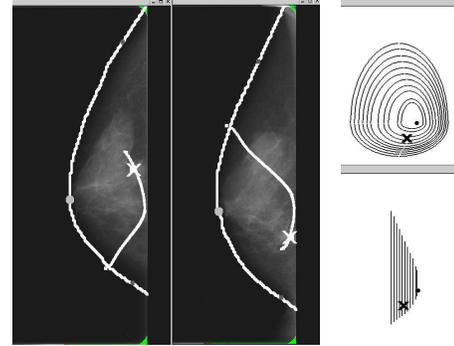


Fig. 2. 3D position of lesion obtained by the CAD system

of the breast.

The breast deformation caused by compression and uncompression is simulated using the simplified geometrical compression model proposed in [16]. This model enables the calculation of the position of any point of the breast under compression from its original position in the uncompressed state, and vice versa as shown in Fig. 1(c). The model was derived based on some approximations on the breast deformation under the compression as like:

Approximation 1(A1). The cross-section for compression of the breast is deformed only in the plane by compression. Here, “the cross-section for compression” means the cross-section cut by the plane which is parallel to the compression direction and perpendicular to the chest wall.

Approximation 2(A2). In the mid-plane between the plate and the cassette, there is no deformation.

Through experiments using about 50 pairs of CC and MLO images of both English and Japanese women [16][1], average error, which is distance from the actual corresponding point to the predicted curve, is less than 7 mm. It overcomes the current radiologists predictions, average error of which is about 10 mm. The accuracy can be improved by replacing

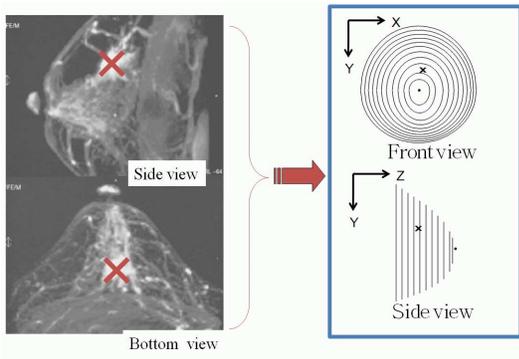


Fig. 3. 3D model reconstructed from MR MIP data

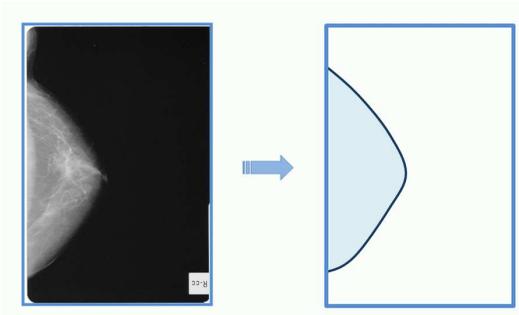


Fig. 4. CC data of the breast of the MR data in Fig.3

the compression model with a richer compression model, Actually, Zhang et al[13] uses a breast FEM model which is reconstructed from MRI data of the same breast to calculate the similar corresponding lines. The accuracy of the experimental results using six patients was reported as about 2.2 mm. However, it should be noticed the former method has merits of working only with the two view mammograms in real time, while the latter requires MRI of the same breast and takes much longer time. This characteristics is important for a handy and easy-to-use CAD system for aiding the diagnosis of mass screening with mammograms only.

Therefore, the improvement of the simplified compression model only with the input images is also desirable. Yam et al [17] improved the compression model by introducing some variable parameters into the model and by adjusting the values based on the correspondences of prominent features (micro-calcifications) on the images. However, mammographic views do not always have enough numbers of corresponding pairs of prominent features.

One direction is to build a more sophisticated geometrical compression model by understanding the breast deformation by compression in more details.

III. SIMULATION OF CC MAMMOGRAPHIC COMPRESSION USING A MECHANICAL MODEL

To understand better the breast deformation by compression, we simulated CC mammographic compression of breast using a mechanical model reconstructed from MR data and compared the results with actual CC image. We use the

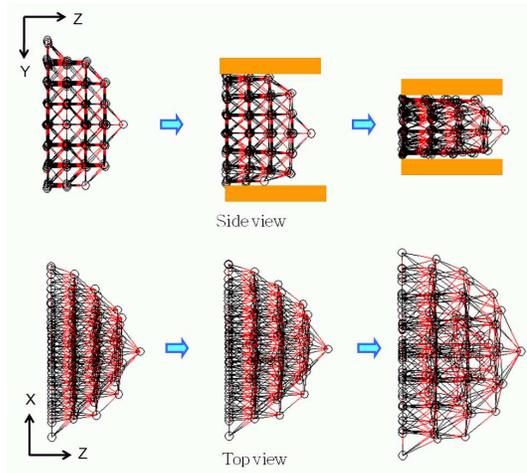


Fig. 5. Compression simulation 1

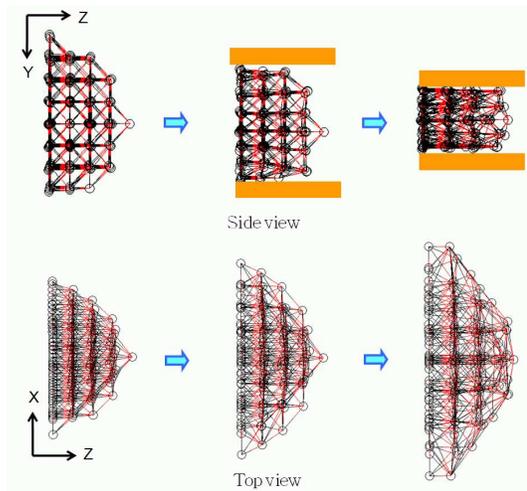


Fig. 6. Compression simulation 2

rheological model developed by Kimura et al.[18], which can simulate the deformation of elastic, viscoelastic, and plastic objects in almost real time. Although the parameters of this model are not directly related with actual biomechanical parameters like Young's modulus, these can be adjusted relatively intuitively. The 3D geometry of the model is reconstructed from outlines in MRI MIP (Maximum Intensity Projection) images as shown in Fig.3. For simulating compression by two plates, Dirichlet boundary conditions (displacement-controlled conditions) are applied to the nodes touched to the plates which move gradually towards each other until the distance between the plates becomes the width at taking the CC image. At the connection to the fixed chest wall, only the z coordinates of the nodes at the chest wall is fixed. Fig. 5 shows a top and side viewsof a simulation result. Although several simulations were tried while manually adjusting the parameters of the viscoelasticity of the model, the outline of the deformed breast did not get close to the one observed in the actual CC image in Fig. 4. The biggest difference is the tendency of elongation in the X

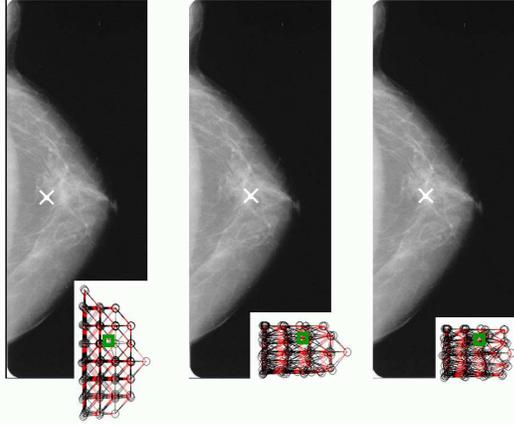


Fig. 7. Projected positions of a lesion detected in MR data

direction, which is perpendicular to the compression and along the chest wall. Although there are also some studies simulating the CC compression[10][13], the consistency of the simulated shape with the actual observation has not been sufficiently investigated. As far as we review, such tendency does not seem in any simulation results. We infer that this is mainly because both the connection with the chest wall and the compression action are fairly complicated and are not well implemented in the simulations. Actually, it is hard to represent this complicated boundary conditions. Instead of sticking to that, we add one more Dirichlet boundary conditions for the nodes on the horizontal plane through the nipple, so as to the outline of the shape after the simulation coincides with the one observed in actual CC image. Concretely, the nodes are forced to move to the position which produce the outline of CC. Fig. 6 shows the deformation of the breast using this condition.

To examine the effects of this modification on the internal deformation, we check the consistency of projected position of a lesion. In Fig. 7, white crosses illustrate three projections of a lesion which is detected in MRI (crosses in Fig. 3): from left to right, the projection of the location at original MRI shape, that after the compression simulation shown in Fig. 5 and that after the compression simulation shown in Fig. 6. As you see, the last one gives best coincidence. This result encourages us to make good use of the features observed in images to specify the deformation.

This observation also gives a clue to the improvement of the geometric compression model used in the CAD system: the modification realizing this tendency of elongation along the chest wall should be added.

IV. OBSERVATION OF INTERNAL DEFORMATION USING DEvised PHANTOM

In order to grasp internal deformation of flexible objects in details, Tokumoto et al.[19] has devised gel phantoms in which small metallic elements arranged with regular intervals and measured the movements of the elements using an industrial CT scanner, TOSCANER-24200AV. Fig.8 shows one example of the experimental results using a semi-ellipsoid

phantom made of human-skin gel. The size of the phantom is 130 (major axis) \times 110 (minor axis) \times 70 (height) mm. 286 metallic elements are arranged inside of the phantom. To observe the internal deformation, the movement of the elements were observed by the CT scanner at three times: with no compression, under compression to the width of 100 mm, and under compression to the width of 80 mm. In Fig. 8(a), white, red and blue colors illustrate the position of the elements at each state respectively.

To observe more clearly, the movement of the metallic elements which on a horizontal plane at the initial state is picked up in Fig. 8(b). The movement in the compression direction (in the Y direction) is big at the part which are pressed directly by the plates (at the part of $X = 60 \sim 100$ in Fig.8(b)). On the other hand, at the remaining outer part, the movement in the X direction, which is perpendicular to the compression direction is stronger. One more noteworthy point is that the former part obviously got denser than the latter part under the compression. This gives us a valuable lesson that too strong volume conservation constraints could cause removal from reality.

This observation suggested some amendments to the compression model used in the CAD system.

- 1) On Approximation 1:
In the current system, for simplification, we ignore the movement perpendicular to the cross-section and consider the deformation only within each cross-section. However, cross-sections tend to bend outward with the biggest displacement at the mid-plane between the plates.
- 2) On Approximation 2:
In the mid-plane between the plates, deformation in the compression direction can be ignored. However, the displacements in the other directions should be considered.
- 3) On 3D reconstruction from the outlines of mammograms:
In the current system, individual 3D breast shape is reconstructed on the assumption that its horizontal and vertical outlines can be approximated with 10-percent scale-downed outlines of CC and MLO images respectively[1]. However, it looks better to take into consideration the distortion of the outlines during the compression rather than assuming the change as similar transformation.

V. CONCLUSIONS

In this paper, we discussed about the model which can simulates breast compression for practical medical applications. To shed light upon the physical deformation of breast under mammographic compression, we have done two-types of experiments: simulation of mammographic compression using a mechanical model and inspection of internal deformation using a devised phantom. Based on the experimental results, Some key issues to improve the compression model used currently in the CAD system[1] were specified in Section IV.

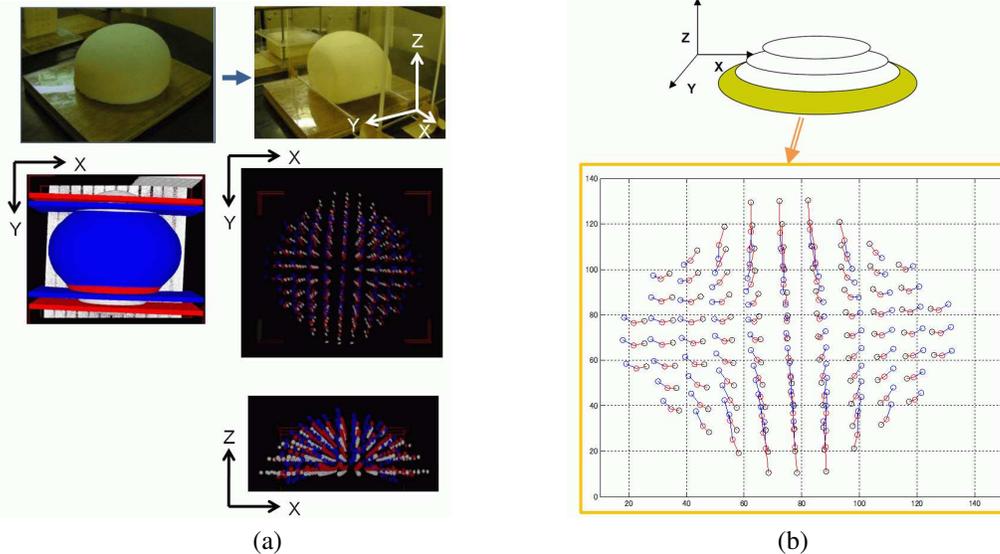


Fig. 8. Inspection results of internal deformation of a phantom with metallic elements inside

Our future work will focus on:

1. Further investigation of the breast deformation under mammographic compression by increasing the number of the experiments in-line with ones shown in this paper.
2. Development of more sophisticated compression model for the CAD system based on the facts obtained from the experiments above.

Acknowledgements

The authors gratefully acknowledge discussions in Medical Vision laboratory, University of Oxford, which has inspired the basic idea of this paper. This research was supported by Grant in Aid for Scientific Research (17206024) by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

REFERENCES

- [1] Y. Kita, E. Tohno, R. P. Highnam and J. M. Brady: "A CAD system for the 3D location of lesions in mammograms", *Medical Image Analysis*, Vol. 6, pp. 267–273, 2002.
- [2] Rueckert, D. and Sonoda, L.I. and Hayes, C. and Hill, D.L.G. and Leach, M.O. and Hawkes, D.J.: "Nonrigid registration using free-form deformations: Application to breast MR images", *IEEE Trans. on Medical Imaging*, Vol. 18, No. 8, pp. 712–721, 1999.
- [3] F.J.P. Richard and L.D. Cohen.: "A New Image Registration Technique with Free Boundary Constraints: Application to Mammography", *Computer Vision and Image Understanding*, Vol. 89, No.2-3, pp. 166–196, 2003.
- [4] C. P. Behrenbruch, K. Marias, P. A. Armitage, M. Yam, N. Moore, R. E. English, J. Clarke, and J. M. Brady: "Fusion of contrast-enhanced breast MR and mammographic imaging data", *Medical Image Analysis*, Vol. 7, pp. 311–340, 2003.
- [5] C. Tanner, J.A. Schnabel, A. Degenhard, A.D. Castellano-Smith, C. Hayes, M.O. Leach, D.R. Hose, D.L.G. Hill and D.J. Hawkes: "Validation of Volume-Preserving Non-Rigid Registration: Application to Contrast-Enhanced MR-Mammography", In *Proc. of 5th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 307–314, 2002.
- [6] Rohlfing, T. and Maurer, Jr., C.R. and Bluemke, D.A. and Jacobs, M.A.: "Volume-preserving nonrigid registration of MR breast images using free-form deformation with an incompressibility constraint", *IEEE Trans. on Medical Imaging*, Vol. 22, No. 6, pp. 730–741, 2003.
- [7] A. Samani, J. Bishop, M. J. Yaffe, and D. B. Plewes: "Biomechanical 3-D Finite Element Modeling of the Human Breast Using MRI Data", *IEEE Trans. on Medical Imaging*, Vol. 20, No. 4, pp. 271–279, 2001.
- [8] F.S. Azar, D.N. Metaxas, and M.D. Schnall: "Methods for modeling and predicting mechanical deformations of the breast under external perturbations", *Medical Image Analysis*, Vol. 6, No. 1, pp. 1–27, 2002.
- [9] Y. Liu, A. E. Kerdok, and R. D. Howe: "A nonlinear finite element model of soft tissue indentation", In *Proc. of Second International Symposium on Medical Simulation*, pp. 67–76, 2004.
- [10] P. Pathmanathan, D. Gavaghan, J. Whiteley, M. Brady, M. Nash, P. Nielsen, V. Rajagopal: "Predicting Tumour Location by Simulating Large Deformations of the Breast Using a 3D Finite Element Model and Nonlinear Elasticity", In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2004*, pp. 217–224, 2004.
- [11] T.J. Carter, C. Tanner, W.R. Crum, and D.J. Hawkes: "Biomechanical Model Initialized Non-rigid Registration for Image-Guided Breast Surgery", In *Proc. of MICCAI 2006 Workshop on Computational Biomechanics for Medicine*, pp. 104–112, 2006.
- [12] V. Rajagopal, J.H. Chung, R. Warren, R. Highnam, M. Nash, and P. Nielsen: "Finite Element Modelling of Breast Biomechanics: Predicting the Effects of Gravity", In *Proc. of MICCAI 2006 Workshop on Computational Biomechanics for Medicine*, pp. 94–103, 2006.
- [13] Y. Zang, Y. Qiu, D. B. Goldgof, S. Sarkar, and L. Li: "3D Finite Element Modeling of Nonrigid Breast Deformation for Feature Registration in X-ray and MR Images", In *Proc. of IEEE Workshop on Applications of Computer Vision*, pp. 38–43, 2007.
- [14] E. Roebuck: "Clinical radiology of the breast", Heinemann medical books, Oxford, 1990.
- [15] R. Novak: "The transformation of the female breast during compression at mammography with special reference to the importance for localization of a lesion", *ACTA radiologica supplement 371* Stockholm, 1989.
- [16] Y. Kita, R. Highnam and M. Brady: "Correspondence between different view breast X rays using curved epipolar lines", *Computer Vision and Image Understanding*, Vol. 83, pp. 38–55, 2001.
- [17] M. Yam, M. Brady, R. Highnam, C. Behrenbruch, R. English, and Y. Kita: "Three-dimensional reconstruction of microcalcification clusters from two mammographic views", *IEEE Trans. Medical Imaging*, vol. 20 no. 6, pp. 479–489, 2001.
- [18] M. Kimura, Y. Sugiyama, S. Tomokuni, and S. Hirai: "Constructing Rheologically Deformable Virtual Objects", In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA2003)*, pp. 3737–3743, 2003.
- [19] S. Tokumoto and S. Hirai: "Internal Deformation Measurements of Flexible Objects for Identification of Parameters by CT Scanner", In *Proc. of SICE SI2006*, pp. 73–74, 2006 [In Japanese].

Modeling, Identification, and Control of Deformable Soft Objects

Control of Deformable Soft Objects

Presentation 1: *Toward Real-time Volume-based Haptic Communication
with Realistic Sensation*

Pages: 97–104

Presentation 2: *Control of Loosely Coupled Joint by Soft Actuators via
Deformable Cartilage*

Pages: 105–113

Toward Real-time Volume-based Haptic Communication with Realistic Sensation

Satoshi Yamaguchi, Takahide Tanaka, H.Q.H. Viet,
Yasufumi Takama, Yoshinori Tsujino, Hiromi T Tanaka
Ritsumeikan University
Computer Vision Laboratory
1-1-1 Noji-higashi, Kusatsu, Japan
yamaguchi@cv.ci.ritsumei.ac.jp

Abstract

A volume-based realistic communication system called Haptic Communication is described. The system allows participants to interact in real time with others at remote locations on the network using haptic perception (sense of touch) of soft objects in virtual environments. We constructed the system so that it provides a sense of touch at remote locations in real time. First, an adaptive volume model represents virtual soft objects in PCs at remote locations. Next, the reflection force of the soft object is calculated rapidly and accurately from the parameters of positions and forces at contacting points transmitted via network at each PC. Eventually, the haptic and visual information are rendered by a haptic device (PHANToM) and a volume graphic software in the PCs. We investigated the efficiency of our system via experiments on a simulation of needle insertion with high force feedback rates at two remote locations on a WAN between Ritsumeikan University, Biwako Kusatsu Campus and Shiga Medical University. The experiment results show that the delay due to network traffic is negligible.

1. Introduction

Virtual reality technology has improved as computers and networks have become faster and more powerful. Some tele-communication systems that allow users to work at remote locations on the network have been developed, and medical and educational applications are expected. This kind of tele-communication we requires visual, auditory and tactile systems. Video chatting is now common. Moreover, force feedback is a tactile way of interacting with virtual objects. It is generated by haptic devices such as PHANToM[10] that can update at 1000 Hz (more than 1000

Hz is generally needed as update rate to achieve a real haptic sensation[3, 4]). A high-speed network library developed to transmit force feedback data through a network at 1000 Hz. Thus, the next generation of tele-communication systems need to have the sense of touch.

The VizGrid project[1] is a part of the IT-Program, a five year Japanese national project initiated in 2002. A goal of the project is to create a real-world oriented remote collaboration environment on a grid based on a volume communication concept. The project has enabled the development of volume modeling of output data from simulation or experimental results and a volume modeling of images in the real world by using a multiple-view camera.

Hikichi et al.[7] proposed a haptic collaboration system without loss of quality of service (QoS). They conducted an experiment to evaluate their system using a rigid and a surface object, and a delay time, packet loss, and information loss were measured.

Mortensen et al.[9] presented a study of remote collaboration between people in a shared virtual environment. Two people, one at University College London (UCL) and the other at University of North Carolina Chapel Hill (UNC-CH), met together in shared virtual environment and lifted a rigid body object together and moved it to another place.

Gunn et al.[6] proposed techniques allowing long-distance sharing of haptic-enabled, dynamic scenes. At the CSIRO Virtual Environments Laboratory, They have successfully used their system to connect prototype of a surgical-simulation application between participants on opposite sides of the world in Sweden and Australia, over a standard Internet connection.

However, previous works did not achieve realistic sensations for the representational model or achieve, real-time performance, and the sense of touch was not well defined. In this study, we constructed our system as follows. First, an adaptive volume model represents virtual soft objects in

the PCs at remote locations. Next, each PC calculates the reflection force of the soft object is rapidly and accurately from the parameters of positions and forces at contacting points transmitted via network. Eventually, the haptic and visual information are rendered by a haptic device such as PHANToM and volume graphic software in the PCs. We tested the efficiency of our system via experiments on simulating a needle insertion with high haptic rendering rates at two remote locations on a WAN between Ritsumeikan University, Biwako Kusatsu Campus and Shiga Medical University. The experimental results show that the delay due to network traffic is negligible. Figure 1 presents an overview of the volume-based realistic communication.

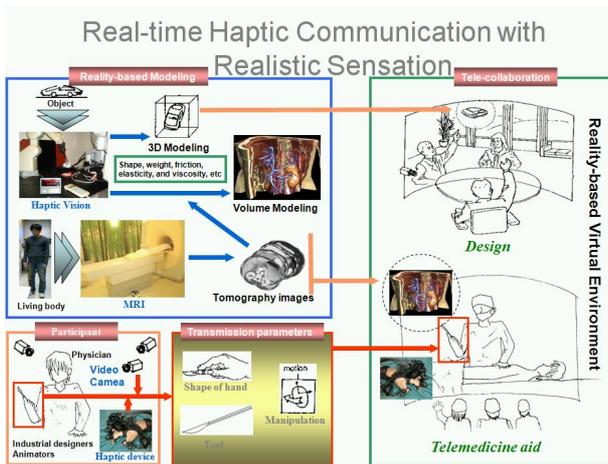


Figure 1. Overview of volume-based realistic communication

2. System Overview and Architecture

2.1. System Overview

In conventional haptic communication systems, a server-client is generally used for the surface model, and all data sets such as image information and force information are transmitted[1]. However, transmitting all data sets of volume models for visualizations is difficult because the volume model is huge. We performed a deformation simulation, shown in Figure 2, using the same volume model at a remote location and minimum manipulation parameters to achieve real-time communication.

2.2. Hardware Architecture

We used a desktop PC on the server side and a laptop PC on the client side. Table 1 has the specification of both sys-

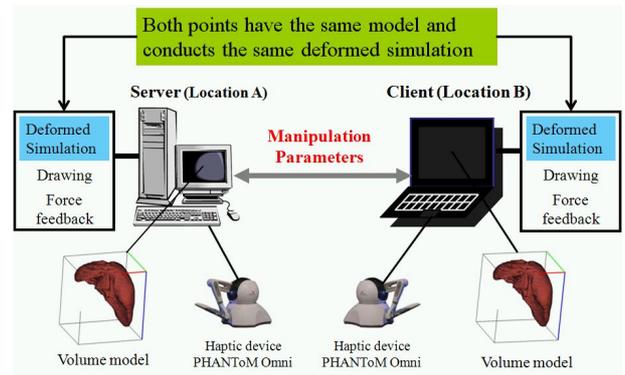


Figure 2. Architecture of system

tems. PHANToM Omni[10] is a haptic device that is connected to both PCs via the IEEE1394 interface. The haptic device enables interaction using the sense of touch a virtual environment with a stylus, which is man-machine interface shaped like a pen.

Table 1. Specifications of the PCs

	OS	CPU[GHz]	RAM[GB]
Server	Windows XP	Intel(R)	2.00
	SP2	Core2 Duo 3.40	
Client	Windows XP	Intel(R)	2.98
	SP2	Core2 Duo 2.33	

2.3. Software Architecture

We used OpenHaptics ToolkitTM[11] for the force feedback. This application enables software developers to add haptic and true 3D navigation to a broad range of applications, from design to games and entertainment to simulation and visualization.

We also used the Haptic Communication Toolkit (HCT)[5] for real-time haptic communication. The HCT is a developer's kit for communication control and also a network library that is developed for communication between two or more haptic devices. However, the HCT can only transmit to 256 bytes to achieve the haptic rate (more than 1000 Hz). To go beyond the limitation is difficult because we need a communication band that is faster and proportional to the size of the transmit data. In this study, we achieved real-time haptic tele-communication using the minimum manipulation parameters.

2.4. Network Architecture

A server-client type is used in our system. The communication band is 1 giga bit per second on the LAN, and the communication protocol is TCP/IP.

3. Deformation Simulation Model

3.1. Mass-Spring-Damper Model

The mass-spring-damper (MSD) models called the Voigt model comprising mass node, springs and dampers for representing deformation are widely applied because of the effectiveness in computation and the simplicity in implementation. Figure 3 shows one of an element in the Voigt model. A tetrahedron consists of the six elements and a set of tetrahedron represents a volume model.

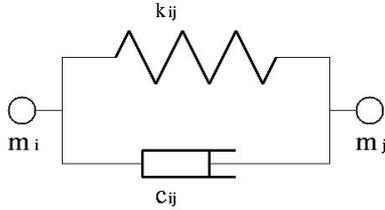


Figure 3. Voigt model

Given a current length l_{ij} , a initial length L_{ij} , and a directional vector of an edge for i th node, the force value F_i is given as follows:

$$F_i = -\sum_j (k(l_{ij} - L_{ij}) + cl_{ij})e_{ij} \quad (1)$$

where k is the stiffness value and c is the damping coefficient. Given a mass of the i th node, a position of the node, and an external force, equations of motion is given as follows:

$$M_i \ddot{P}_i = F_i + F_i^{ext} \quad (2)$$

3.2. Collision Detection

An algorithm to detect a collision between a stylus of a haptic device and a node is required for manipulation of a visco-elastic model using the PHANTOM on virtual environment. Let $\vec{OA} = (a_1, a_2, a_3)$, $\vec{OB} = (b_1, b_2, b_3)$, $\vec{OC} = (c_1, c_2, c_3)$ be vectors comprising a tetrahedron (Figure 4), A volume of the tetrahedron is given as follows:

$$V = \frac{1}{6} |a_1b_2c_3 + a_2b_3c_1 + a_3b_1c_2 - a_1b_3c_2 - a_3b_2c_1 - a_2b_1c_3| \quad (3)$$

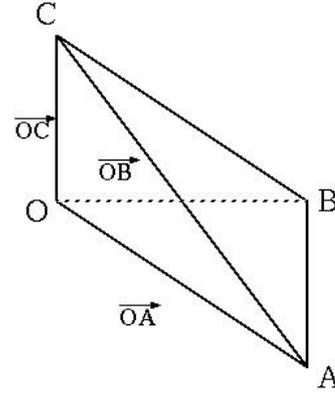


Figure 4. Tetrahedron

As shown Figure 5, a position P_i of the stylus newly generate four tetrahedrons. The volume of generated tetrahedron is let by Equation 3. If a sum of all generated tetrahedron equals to the volume of initial tetrahedron, the collision is detected.

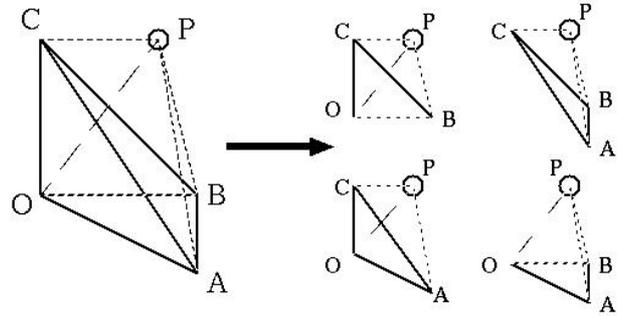
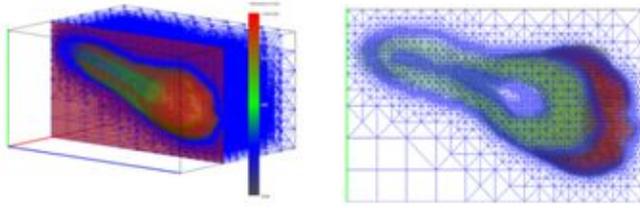


Figure 5. Devided tetrahedrons

4. Adaptive Volume Modelling

In our previous work[12], we developed a deformation simulation of a visco-elastic object using the mass-spring dumper model. An input mesh model such as an organ is represented using a binary tree of a set of tetrahedrons without any cracks forming. The model is based on a tetrahedral adaptive mesh for parallel hierarchical tetrahedralization of volume data (Figure 6).

The interaction with the virtual object, coordinate, velocity, acceleration, and mass were set to all nodes of the tetrahedral adaptive mesh to simulate the deformation.



(a) Tetrahedral adaptive mesh (b) Cross-sectional view

Figure 6. Tetrahedral adaptive mesh (Volume model of tooth)

The visco-elastic parameters, elastic parameters and initial length of the edge were set to the edge, consisting of two nodes.

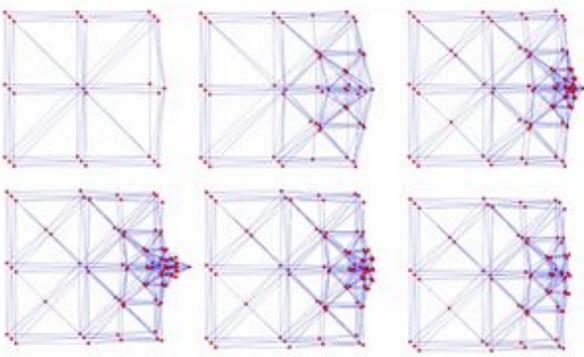


Figure 7. Dynamic tetrahedral adaptive grids of volume data

Furthermore, we used the dynamic tetrahedral adaptive grid of volume data[13] to simulate deformation of the visco-elastic object rapidly. This algorithm can refine above tetrahedral adaptive mesh interactively. The tetrahedral adaptive mesh is locally subdivided based on the expansion and contraction rate at each edge. The object is constantly represented by specified approximate precision. Figure 7 shows the dynamic tetrahedral adaptive grids of volume data and how they progress. This paper features an extension of this model for manipulation using the haptic device.

5. Haptic Communication

5.1. Send/Receive Packet

The manipulation parameters for haptic communication using the deformation simulation are as follows.

- 3D position of a tip of the stylus in virtual space ($8[byte] \times 3 = 24[byte]$)
- Tetrahedron ID held by users ($8[byte]$)
- Node ID manipulated by users ($8[byte]$)
- 3D position of node manipulated by users ($8[byte] \times 3 = 24[byte]$)

The manipulation parameters are 64 bytes less than the limitation of the HCT, 256 bytes.

5.2. Process Flow

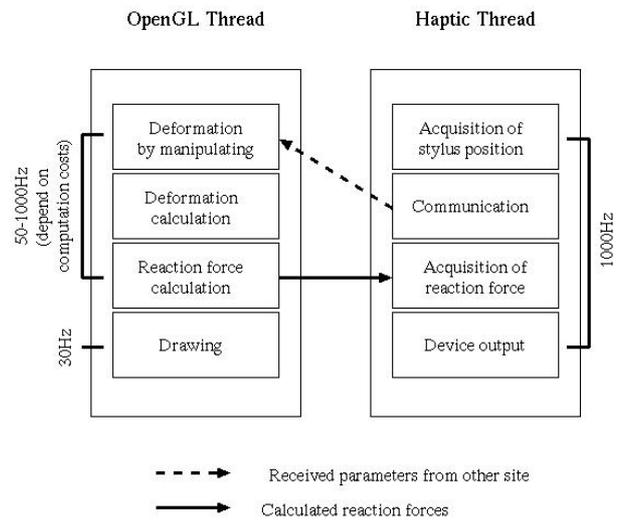


Figure 8. Process flow

As shown Figure 8, our system has two threads. One is a OpenGL thread and the other is a Haptic thread. the OpenGL thread consists of four processes, a deformation by manipulation, a deformation calculation, a reaction force calculation, and a drawing. An update rate of the OpenGL thread depends on a resolution of the volume model. In this study, a cube of initial level 3 (a number of initial node is 24) is used. This level changes to level 9 (a number of node is 729) by deformation. Thus, the update rate ranges from 50 Hz to 1000 Hz with changing resolution of the volume model. The update rate of the drawing process is controlled as 30 Hz.

The haptic thread also consists of four processes, an acquisition of the stylus position, a communication, an acquisition of the reaction force, and a device output. The update rate of the haptic thread is 1000 Hz.

6. Experiment

The previous haptic tele-communication system experienced, time delays caused by network traffic and reduced accuracy of force feedback [8]. We performed an experiment on a WAN to evaluate the time delay and force feedback accuracy in our system.

6.1. Experimental Condition

We performed an experiment using two remote locations on a WAN between Ritsumeikan University, Biwako-Kusatsu Campus (RU-BKC), and Shiga Medical University (SMU). The distance between the universities on a straight line is about 2 kilometers. We also performed same experiment between RU-BKC and Osaka University, Toyonaka Campus (OU-TC). The distance between the universities on a straight line is about 52 kilometers. Furthermore, we performed an experiment using three remote locations on a WAN and a LAN. One connection between RU-BKC and OU-TC was used the WAN, the other connection between OU-TC and OU-TC was used the LAN. The communication time was measured based on the following conditions.

- Measure the round trip time between server and client
- Measure the number of frames from 10,000 to 25,000 (only correct frames)
- Calculate average times, maximum times and minimum times
- Repeat five times
- Measure the round trip time between server and client (not between client and client) in the case of the tele-communication at three remote locations

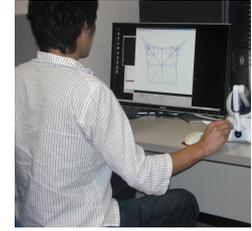
6.2. Experimental Results

Two users performed an interaction of soft objects in a virtual environment. Figure 9 shows a scene of an experiment by two users and Figure 10 shows each displayed models. Figure 10(a) shows the server side interaction and Figure 10(b) shows the client side interaction. Figure 10 shows a representation of the deformation that arises by the interaction.

Table 2, Figure 11 (a) and Figure 11 (b) show the experimental results between RU-BKC and SMU. Table 2 shows a value of the round trip times of all trials. Figure 11 (a) shows the round trip time on the 1st trial. Figure 11 (b) shows the maximum times, minimum times and average times of all trials.

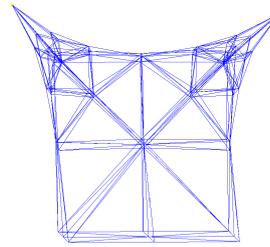


(a) Server (Location A)

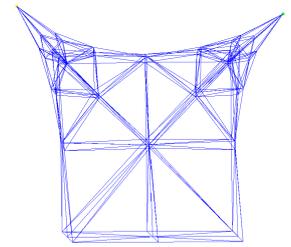


(b) Client (Location B)

Figure 9. Haptic telecommunication by two users (Scene)



(a) Server



(b) Client

Figure 10. Haptic telecommunication by two users (Display)

As shown in Table 2, the average times of all trials are about 11,000 micro seconds. The maximum time of all the trials is 30,139 micro seconds.

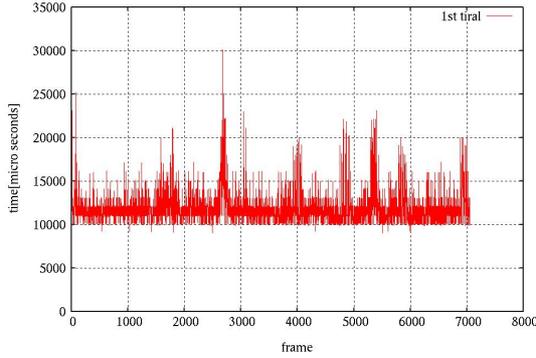
Table 3, Figure 12 (a) and Figure 12 (b) show the experimental results between RU-BKC and OU-TC. Table 3 shows a value of the round trip times of all trials. Figure 12 (a) shows the round trip time on the 1st trial. Figure 12 (b) shows the maximum times, minimum times and average times of all trials.

As shown in Table 3, the average times of all trials are about 13,000 micro seconds. The maximum time of all the trials is 268,743 micro seconds.

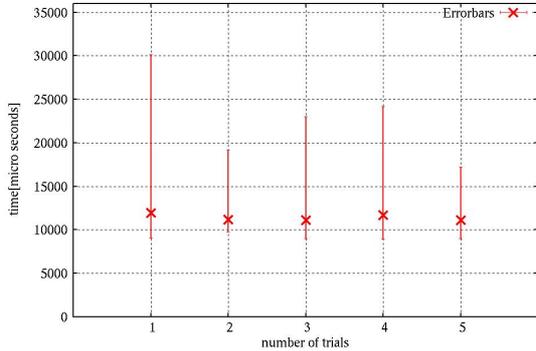
Table 4, Table 5, Figure 13 (a), Figure 13 (b), Figure 13 (c) and Figure 13 show the experimental results in three remote communications (RU-BKC, OU-TC Location A and OU-TC Location B). Table 4 and Table 5 show a value of the round trip times of all trials. Figure 13 (a) and Figure 13 (c) show the round trip time on the 1st trial. Figure 11 (b) and Figure 13 (d) show the maximum times, minimum times and average times of all trials. Table 4, Figure 13 (a) and Figure 13 (b) show the experimental results at OU-TC Location A. Table 5, Figure 13 (c) and Figure 13 (d) show the experimental results at OU-TC Location A.

Table 2. Round trip times of all trials between RU-BKC and SMU [μ sec]

	1st	2nd	3rd	4th	5th
Maximums	30,139	19,148	23,024	24,152	17,211
Minimums	9,044	9,798	9,008	8,964	9,025
Averages	11,943	11,178	11,103	11,685	11,116



(a) Round trip times on the 1st trial



(b) Average times, maximum times and minimum times

Figure 11. Experimental results between RU-BKC and SMU

As shown in Table 4 and Table 5, the average times of all trials are about 15,000 micro seconds. The maximum time of all the trials is 998,508 micro seconds and 1,126,960 micro seconds.

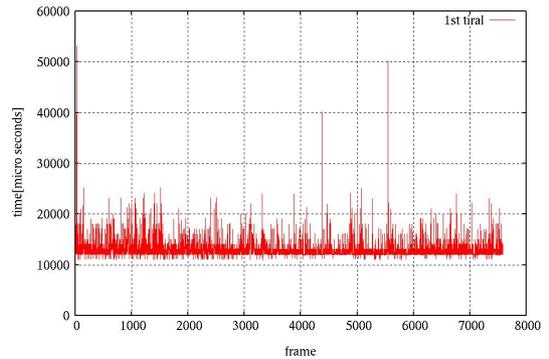
Figure 14 (a) and Figure 14 (b) show the round trip time all days once every hour.

6.3. Discussions

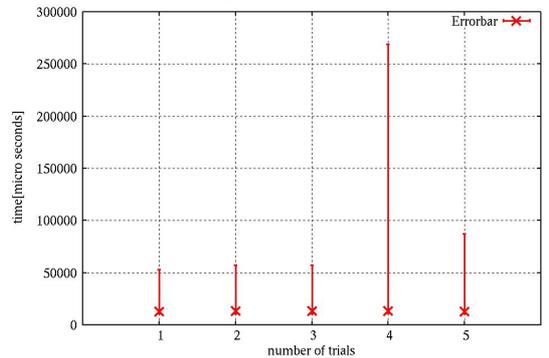
In any cases, the subjects did not feel any discontinuity of force feedback during while the experiment. The difference between the maximum time and the minimum time was caused by a priority of communications on a WAN. As shown Figure 11 (b), Figure 12 (b), Figure 13 (b) and Figure 13 (d), the average times are almost same as the minimum times. In other words, the communication is almost

Table 3. Round trip times of all trials between RU-BKC and OU-TC [μ sec]

	1st	2nd	3rd	4th	5th
Maximums	53,143	57,166	57,153	268,743	87,133
Minimums	10,893	10,891	10,889	11,020	10,893
Averages	13,110	13,186	13,183	13,574	13,120



(a) Round trip times on the 1st trial



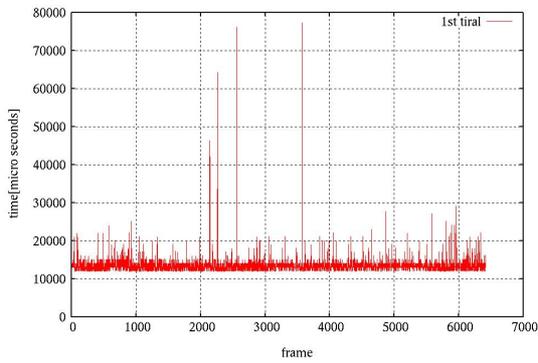
(b) Average times, maximum times and minimum times

Figure 12. Experimental results between RU-BKC and OU-TC

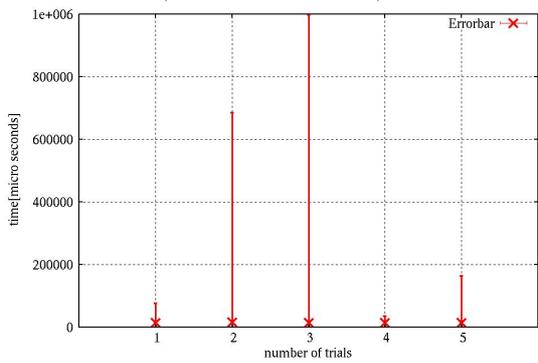
achieved by the minimum time.

All experiment were performed at 10 a.m. As shown Figure 14 (a) and Figure 14 (b), the amount of network traffic at 10 a.m. on each locations is about 13[msec]. This results give suggestion that the communication performance depends on the running time.

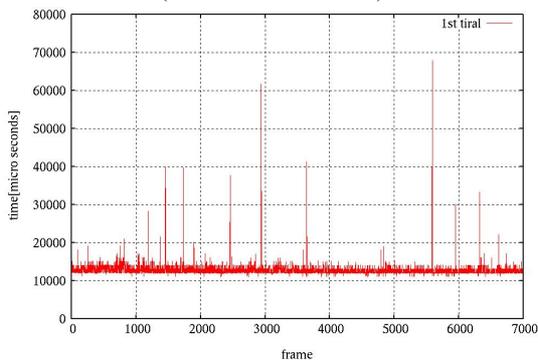
Hikichi et al.[7] proposed a haptic collaboration system without loss of quality of service (QoS). They conducted an experiment to evaluate their system using a rigid and a surface object, and a delay time, packet loss, and information loss were measured. Their results show a maximum allowable limit of the delay time is about 80[msec]. In this study, the experimental results show the round trip time is totally about 10 to 20[msec] less than 80 [msec]. Of course, we used a soft object and volume model by different pa-



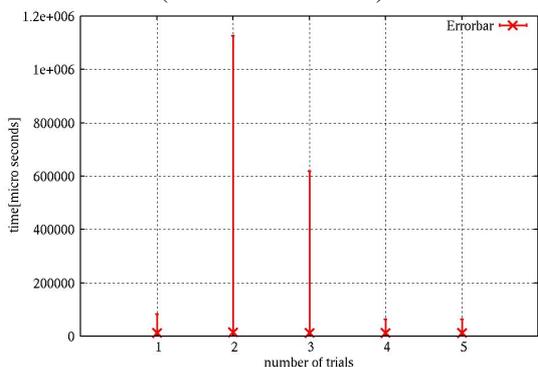
(a) Round trip times on the 1st trial (OU-TC Location A)



(b) Average times, maximum times and minimum times (OU-TC Location A)



(c) Round trip times on the 1st trial (OU-TC Location B)



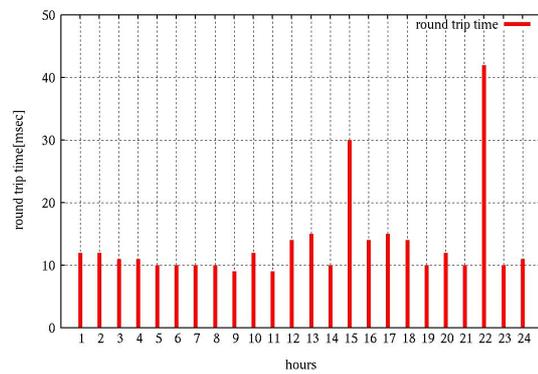
(d) Average times, maximum times and minimum times (OU-TC Location B)

Table 4. Round trip times of all trials (OU-TC Location A)) [μ sec]

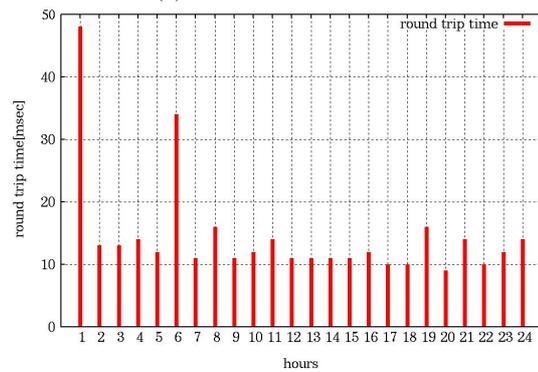
	1st	2nd	3rd	4th	5th
Maximums	77,266	684,837	998,508	36,133	164,391
Minimums	11,886	11,234	11,022	10,999	11,854
Averages	13,448	16,191	14,667	13,978	14,398

Table 5. Round trip times of all trials (OU-TC Location B)) [μ sec]

	1st	2nd	3rd	4th	5th
Maximums	83,307	1,126,960	618,546	65,058	65,206
Minimums	10,987	11,016	10,988	10,880	10,857
Average	12,753	14,699	13,595	13,221	13,479



(a) RU-BKC and SMC



(b) RU-BKC and OU-TC

Figure 14. Round trip time all days once every hour

rameters and different experiment procedures in comparison with their experiment. However, in the case of using the soft object, we don't need high haptic rate such as 1000 Hz[2].

Figure 13. Experimental results in three remote communications (RU-BKC, OU-TC Location A and OU-TC Location B)

7. Conclusions

We described a volume-based haptic communication system that shares an adaptive volume model between remote locations and provides haptic communication to users. Furthermore, we investigated the efficiency of our system via experiments on a simulation of a needle insertion with high haptic rendering rates at two remote locations on a WAN between Ritsumeikan University, Biwako Kusatsu Campus (BKC), and Shiga Medical University and at three remote locations on a WAN between Ritsumeikan University, BKC, and Osaka University, Toyonaka Campus (TC). The experimental results show that the delay due to network traffic is negligible. In future work, we will extend the capability of our system by using multi core CPUs, by synchronizing visualization between the server and client, and by developing an interpolation algorithm for force feedback. We will also perform an experiment by some subjects for subjective assesment.

8. Acknowledgement

This research was funded by the Strategic Information and Communication R&D Promotion Program in 2005(No.051307017) of the Ministry of Internal Affairs and Communications, Japan.

References

- [1] VizGrid Project, Final Report of Vizgrid Project. Technical report, Japan, 2007.
- [2] C. Basdogan, C.-H. Ho, and M. A. Srinivasan. Virtual environments for medical training: Graphical and haptic simulation of laparoscopic common bile duct exploration. *Trans. on Mechatronics*, 6(2):269–285, 2001.
- [3] G. C. Burdea. *Force and Touch Feedback for Virtual Reality*. Wiley-Interscience, 1996.
- [4] G. C. Burdea and P. Coiffet. *Virtual Reality Technology*. IEEE, 2003.
- [5] I. Goncharenko, M. Svinin, S. Matsumoto, Y. Masui, Y. Kanou, and S. Hosoe. Cooperative control with haptic visualization in shared virtual environments. In *Proc. of the 8th Intr. Conf. on Information Visualization*, pages 533–538, 2004.
- [6] C. Gunn, M. Hutchins, and M. Adcock. Combating latency in haptic collaborative virtual environments. *Presence: Teleoperators and Virtual Environments*, 14(3):313–328, 2005.
- [7] K. Hikichi, H. Morino, I. Fukuda, S. Matsumoto, K. Sezaki, and H. Yasuda. Proposal and evaluation of system for haptics collaboration. *Journal of The Institute of Electronics, Information and Communication Engineers*, J86-D(2):268–278, 2003.
- [8] J.Arata, H.Takahiro, P.Pitakwatchara, S.Warisawa, K.Konishi, K.Tanoue, S.Ieiri, S.Shimizu, N.Nakashima, K.Okamura, Y.S.Kim, S.M.Kim, J.S.Hahm, M.Hashizume, and M.Mitsuishi. A remote surgery experiment between japan-korea using the minimally invasive surgical system. In *Proc. of Intr. Conf. on Robotics and Automation*, pages 257–262, 2006.
- [9] J. Mortensen, V. Vinayagamoorthy, M. Slater, A. Steed, B. Lok, and M. Whitton. Collaboration in Tele-Immersive Environments. In *8th Eurographics Workshop on Virtual Environments*, pages 93–101, 2002.
- [10] SensAble Technologies Inc. *Phantom OmniTM User's Guide*, 2004.
- [11] SensAble Technologies Inc. *OpenHapticsTM Toolkit Ver.2.0 Programmer's Guide*, 2005.
- [12] Y. Takama, A. Kimura, and H. T. Tanaka. Tetrahedral adaptive mesh for parallel hierarchical tetrahedralization of volume data. *Journal of The Information Processing Society of Japan*, 48(SIG 9(CVIM 18)):67–78, 2007.
- [13] Y. Takama, H. Yamashita, and H. T. Tanaka. Dynamic tetrahedral adaptive mesh generation of volume data. In *Symposium on VC/GCAD2007*, 2007.

Control of Loosely Coupled Joint by Soft Actuators via Deformable Cartilage

Mizuho Shibata, Takahiro Yoshimura and Shinichi Hirai

Abstract— We introduce a robotic prototype of an arm with a *loosely coupled joint*, modeled on the human joint. A viscoelastic object functions as cartilage and soft actuators as muscles. First, we show that although viscoelastic object affords smooth movement owing to shift in the center of rotation. The prototype was shown to be highly robust against mechanical disturbance owing to its good mechanical compliance. We also describe here a mechanism for controlling angles of a human-like joint using length sensors in three-dimensional (3D) space. In addition, we propose an appropriate method of measurement to reduce errors in measurement due to the length sensors. Using this method, we found that, for each projecting plane, the errors were less than 1.0 deg in our 3D prototype.

I. INTRODUCTION

Tasks performed by a human arm with compliant joints are robust for environmental variations. Humanoid robots require such robust motions to perform actions similar to those of humans. Up to now, researches on mechanisms that realize human joint motion have been studied. Okada *et al.* developed a cybernetic shoulder, which imitates the motion of a human shoulder mechanically [1], [2].

Results obtained with this cybernetic shoulder indicate the need for humanoid robots to have high, human-like mobility and sensitive compliance. The movement of the center of rotation of the cybernetic shoulder yielded the smooth motions of a human shoulder joint using a closed-loop link mechanism with 3-DOF joints. In contrast to the cybernetic shoulder, which consists of rigid links, a human shoulder consists of rigid bones, soft muscles and a soft cartilage between the bones. The cartilage is porous, and the interstitium is filled with fluid. Under stress, fluid moves in and out of the tissue, and the mechanical properties of a cartilage change with fluid movement [3], suggesting that the motion and compliance of a human shoulder joint may be realized by a mechanism with soft components.

Since elastic joints contribute to improving robustness for the environment due to the compliance involved, many studies have analyzed the ability to control a robot with elastic joints [4], [5], [6]. For example, although the viscous terms of a viscoelastic joint may affect the static precision of positioning or the dynamic accuracy of trajectory tracking tasks, these viscous terms contribute to controlling stability [7]. By leveraging the mechanical properties of the joint, the

M. Shibata is with Department of Robotics, Ritsumeikan Univ., Japan
rr003983@se.ritsumei.ac.jp

T. Yoshimura is with Department of Robotics, Ritsumeikan Univ., Japan
rr010031@se.ritsumei.ac.jp

S. Hirai is with Department of Robotics, Ritsumeikan Univ., Japan
hirai@se.ritsumei.ac.jp

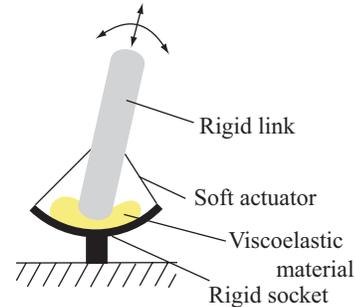


Fig. 1. Concept of a loosely coupled mechanism

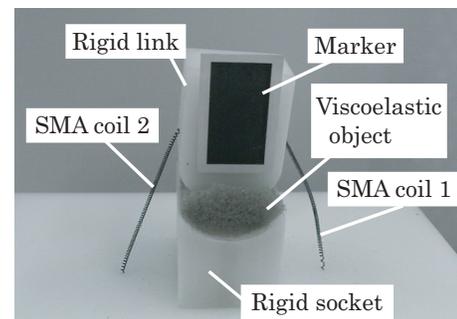


Fig. 2. Prototype of a loosely coupled mechanism

joint mechanism can work stably and robustly under simple control laws.

Researches on the control a link mechanism including viscoelastic objects have been performed. For example, a robot that includes a flexible spine [8], [9], consisting of a series of ball-and-socket joints covered by rubber, can twist its upper body through the spine. Although controlling tendon length, tension, and spring can control the spine, the effect of soft cartilage on robust control of the robot was not determined, and spine compliance was not verified experimentally. The human arm, including cartilage, has been modeled to simulate arm motion, leading to the formulation of a dynamic equation of the human arm, including soft cartilage, by a bond graph simulating arm motion, in which the cartilage was assumed to have a nonlinear spring component and a linear damper component [10].

We propose a novel link mechanism to investigate effects of soft component in control of a human-like joint. Here, we introduce a robotic prototype of an arm with a loosely coupled joint modeled on the human arm, with viscoelastic

material and soft actuators functioning as the cartilage and muscles of a *loosely coupled joint*. We also describe the construction of a robotic hand with high robustness for environmental variation using a mechanism with length sensors. We describe here the control method of a three-dimensional (3D) loosely coupled mechanism using length sensors. After summarizing the concept of the loosely coupled mechanism, we describe the construction of a prototype of this mechanism in two-dimensional (2D) space to confirm motion of the link using a length sensor. Finally, based on the prototype of the 2D mechanism, we constructed a 3D loosely coupled mechanism with length sensors. In addition, we propose an appropriate method of measurement to reduce errors in measurement due to the length sensors.

II. LOOSELY COUPLED JOINT

A. Concept

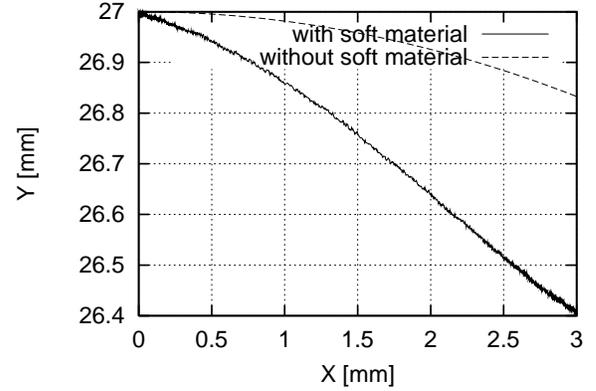
The concept of the loosely coupled joint is depicted in Figure 1. This mechanism consisted of a revolute joint with soft actuators driving a rigid link in a rigid socket containing a viscoelastic object; these correspond to the muscles, bones, sockets and cartilage of a human arm. The soft actuators and viscoelastic object make the joint compliant, like a human joint. Like the muscles, the actuators expand and contract unidirectionally and work antagonistically as a pair to rotate the link. Importantly, because of the viscoelastic object, the center of rotation of the link is not fixed.

B. Specifications and system

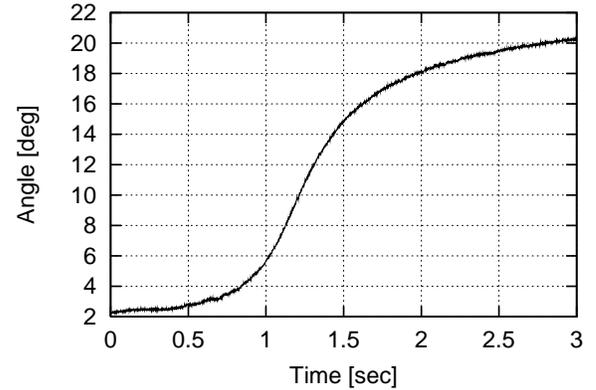
The prototype is shown in Figure 2. The actuators are made from BMX200 shape memory alloy (SMA) coil (TOKI Corporation, Japan), the link and socket are made from polyoxymethylene, and the viscoelastic object is an off-the-shelf sponge, 10 mm in thickness, 0.08 g in weight, and with a Young's modulus of about 20 kPa. To prevent slip, the sponge is fixed to the link and socket. The rigid link and the socket have a combined weight of 21.7 g, and the link end and origin of the joint are separated by a distance of 27 mm. The size of the complete joint mechanism is about that of a human fingertip. The actuators are powered through a ULN2003AN driver (Texas Instruments, USA). With a rectangular piece of black paper (9.5 x 19.5 mm²) on the bottom of the link as a marker, the position and orientation of the link were calculated from the moments of the first and second orders, respectively, of images captured using a 1,000 Hz high-speed camera [11].

C. Shift in center of rotation under open-loop control

When the voltage v_{inp}^1 applied to the driver for one of the actuators was set to 5 V while the voltage v_{inp}^2 to the other actuator was 0 V, X, the position of end of the link in horizontal direction, moved relative to Y, the position of end of the link in vertical direction, as indicated by the solid line in Figure 3-(a). In contrast, when a solid object replaced the viscoelastic object the end of the link moved as indicated by the broken line, which is an arc of a circle centered at the origin of the joint. Clearly, the center of rotation of the



(a) End-point position



(b) Link angle

Fig. 3. Experimental results under open-loop control

link is not fixed, similarly to the human joint, particularly in the presence of the viscoelastic object. Additionally, Figure 3-(b) shows that the link angle increases monotonously as time advances. Therefore, the link can move smoothly, like a human joint.

D. Variable compliance

The compliance of the link mechanism was tested experimentally using the setup shown in Figure 4. The 50 g weight caused the link to move 12 deg when no voltage was applied to the actuators at any time. The combination of the weight and v_{inp}^1 set to 3.0 V and v_{inp}^2 to 2.0 V simultaneously, caused the link to move 7.5 deg. When the thread connecting the weight to the link was severed by burning without changing the voltage applied to the actuators, the angle of the joint, monitored using a 1,000 Hz CMOS camera, changed as shown in Figure 5. The results show that the compliance is higher when a voltage is applied to the actuators and that in this state, as expected, the actuators and the cartilage are stiffer. Additionally, the viscous properties of the link mechanism are little changed when the voltage is applied to the actuators.

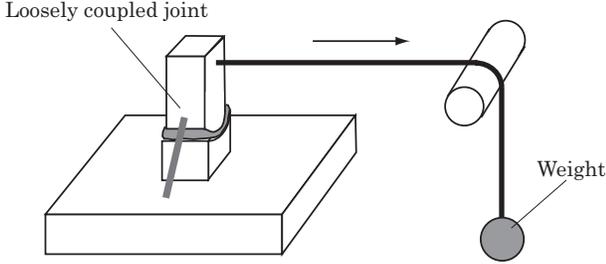


Fig. 4. Experimental setup for measuring compliance

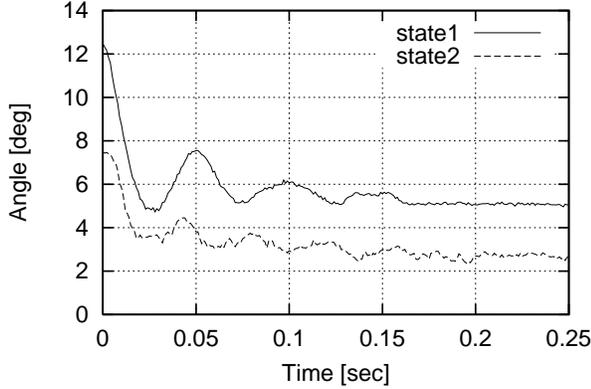


Fig. 5. Test results for compliance

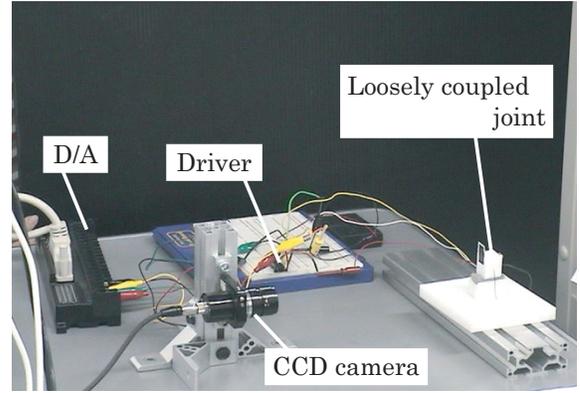


Fig. 6. Control system

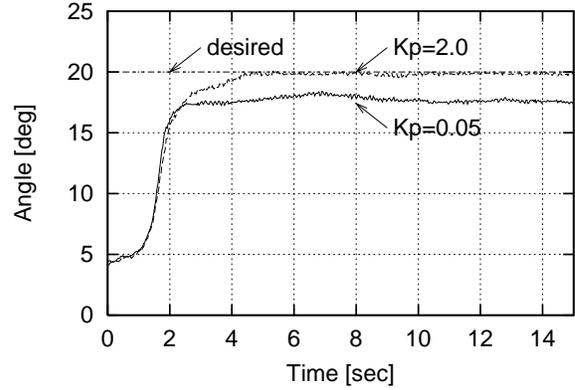


Fig. 7. Experimental results under P control

III. ANGLE CONTROL USING VISUAL FEEDBACK

A potentiometer would not be much help because the center of rotation is not fixed. Therefore, we adopted visual feedback to control the angle of the joint. We used proportion laws that do not depend on knowledge of the physical properties of the joint because, in the case of soft materials, the physical properties, especially the viscosity, are not accurately known. We did not attempt to control the velocity of the joint because it would be more data intense.

A. Experimental setup

Figure III shows the visual feedback system. Two personal computers (PCs) were connected using a 1 Gbps optic fiber (AVAL DATA, Japan). The time lag between the PCs was negligible. One PC was used to generate control inputs to the actuators at a frequency of $10 \mu\text{s}$ and the other was used to store and process the visual information captured by the CCD camera operating at the standard NTSC frame rate of 30 per second. The sampling rate of the sensor was adequate because SMA actuators have slow responses.

B. P control

We used the following simple proportional (P) control law for the link:

$$\begin{cases} v_{inp}^1 = -K_P(\theta(t) - \theta_d) + v_{offset}, \\ v_{inp}^2 = 0, \end{cases} \quad (1)$$

where θ_d is the desired angle, to be held constant, K_P is the proportional gain which is a positive constant, and v_{offset} is the minimum voltage required for the drivers, which was 1.7 V for our drivers. The load on the actuator complicates the relationship between input voltage and the actuator's driving force, hence use of the input voltage to the driver IC as the control input in Eq.1. Figure 7 shows how the angle changes with time when the desired angle θ_d was 20 deg and K_p was 0.05 or 2.0. The angle reaches a steady value, albeit there is some fluctuation. This steady-state error depends on the value of K_P , increasing with gain. However, even when the gain K_p to 2.0, the steady-state error is acceptable, being less than 0.15 deg.

Therefore, we can control the angle of the link, within a positive range, by one actuator. We can explain these results by examining position control of the viscoelastic object fixed on a wall as shown in Figure 8. Let x and f_{drive} be the position of a mass point and the driving force acting on the mass point, respectively. Here, the dynamic equation of the mass point can be expression as $m\ddot{x} = -Kx - B\dot{x} + f_{drive}$, where m , K , and B are the mass, stiffness and viscous coefficients of the mass point, respectively. Let x_d represent the desired position. Applying driving force $f_{drive} = -K_P(x(t) - x_d)$, the mass converges to a certain position

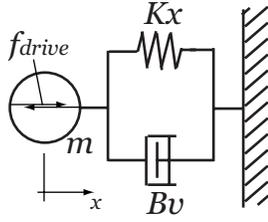


Fig. 8. Viscoelastic object fixed to a wall

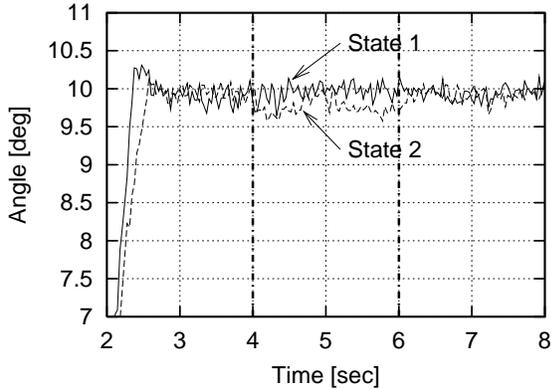


Fig. 9. Test results for robustness against disturbance

with a steady-state error as the gain increases. This matches the experimental results obtained above. The viscosity of a soft material contributes to the stability. Hence, the joint can be controlled by feedback control without damping. Now, let K_I be an Integral gain and be a positive constant. Using Integral (I) control law, that is, applying the driving force $f_{drive} = -K_I \int_0^t (x(t) - x_d) dt$, the mass converges to the desired position correctly. Hence, we think that the joint will converge to a desired location correctly under I control only. However, we have not verified it experimentally because the actuators have a slow response, but we aim so study it in the future work.

C. Robustness under P control

We experimentally investigate the robustness of the joint for disturbances. We used one SMA actuator as a disturbance, and compare two states. In state 1, we continuously applied the v_{inp}^1 and v_{inp}^2 as in Eq. 1. In state 2, we set $v_{inp}^2 = 5.0$ V from 4.0 s to 6.0 s, making actuator 2 a disturbance for SMA actuator 1. Figure 9 shows how the angles changed in the two states, with $K_p = 2.0$ and $\theta_d = 5.0$ deg. The angle converges to a stable location after v_{inp}^2 breaks contact. We experimentally confirm that the disturbance by SMA actuator 2 is about 150 g weights. The weights generate a momentum for the rigid link as much as the momentum applied by actuator 1, and it is a large disturbance for the joint. These results imply that P control of the joint is robust.

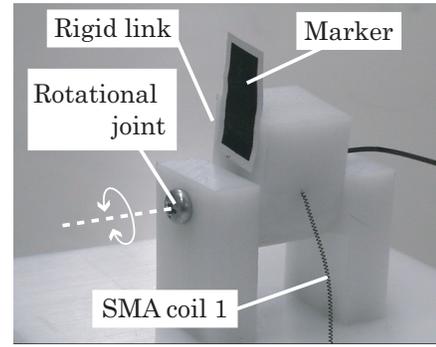


Fig. 10. Joint mechanism without the cartilaginous area

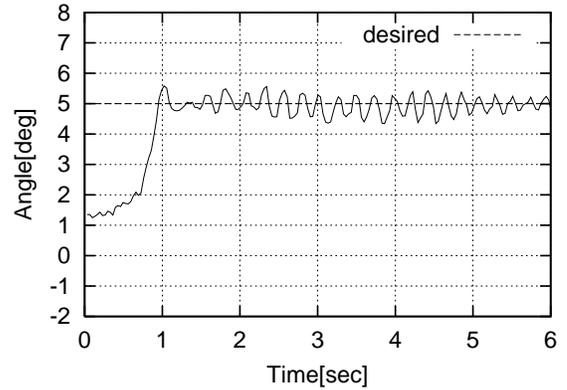


Fig. 11. Experimental result of joint without the cartilaginous area

D. Comparison with Normal link

As described in the previous section, we made a link mechanism with a cartilaginous area. In this section, we investigate the availability of the link mechanism. Figure 10 shows the link mechanism without a cartilaginous area. The link mechanism has a rotational joint with a bearing. In addition, the trajectory of the rigid link describes an arc since the center of rotation is fixed. We arrange a pair of SMA actuators to rotate the rigid link. The size of the link mechanism is the almost same as the size of the link with loosely coupled joint shown in Figure 2. Figure 11 shows an experimental result of P control in Eq.1. In this control, the actuator 2 does work instead of a bias spring which generates a restoring force for shrinkage of a SMA actuator since the SMA actuators are arranged antagonistically. By comparing of Figures 7 and 11, it is clear that the viscoelastic object reduces the oscillation in the angle.

IV. ANGLE CONTROL USING LENGTH SENSORS

As described in the previous section, we used visual feedback to realize an angle control of the link mechanism. Because the joint has high compliance and damping characteristic under P control, the motion of the joint was highly robust against disturbances. The motion of the joint mechanism, however, was lowly robust when occlusion occurred during visual feedback. Human muscles have spindles,

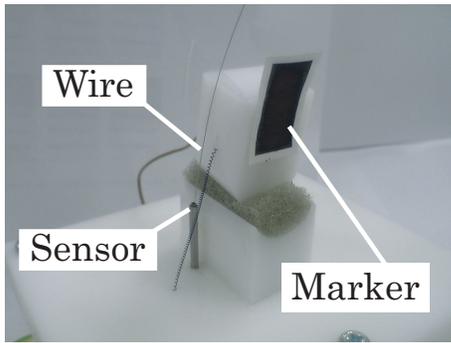


Fig. 12. Loosely coupled mechanism with one wire sensor

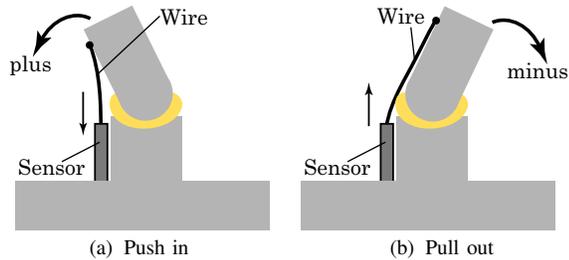


Fig. 13. Sensor configuration for a 2D type loosely coupled mechanism

allowing the current length of the muscles to be measured [12]. Thus, humans can use not only visual information about an object but length information of their muscles to move their arms. Length information for angle control improves robustness, especially, in 3D space. We therefore adopted a sensor that could measure length directly.

V. ANGLE CONTROL USING A LENGTH SENSOR IN 2D SPACE

A. Angle measurement

An electric resistance feedback method has been used to measure a length of SMA actuator [13]; in this method, the length of the SMA actuator was estimated by the electric resistance variation. Although this scheme has been applied to controlling an active endoscope, this method requires SMA actuators in a constant loaded condition. Since the loads on SMA actuators vary during control using this mechanism, we used a pulse coder (LEVEX, Japan) as a length sensor. Figure 12 shows a prototype of a 2D loosely coupled mechanism with one length sensor. This sensor can measure the length of SUS 304 wire (diameter: 0.2 mm) inserted into a sensor tube. The sensor is arranged along one side of the socket (Figure 13). When the angle of the link increases, the wire pushes into the tube from an offset position; when the angle decreases, the wire pulls out the tube from an offset position. To determine the relationship between link angle and sensor length, we calibrated the link angles by CCD images, and the orientation of the link was calculated from the second order momentum of the rectangular black marker (Figure 12). Figure 14 shows the angle identification for the length sensor. Using an insert of wire d , a link angle θ could be

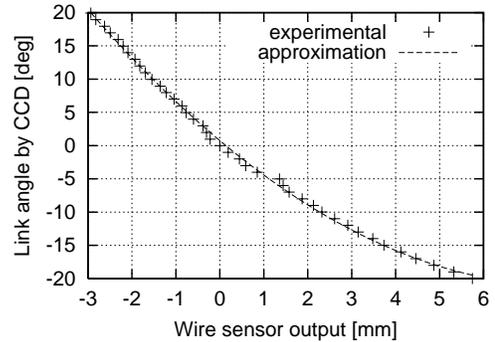


Fig. 14. Angle identification of a length sensor (2D)

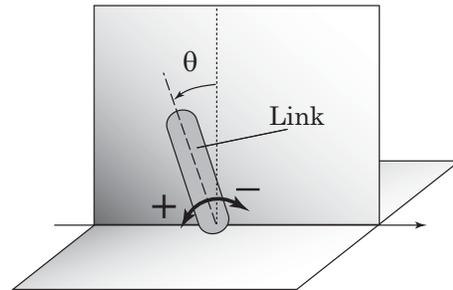


Fig. 15. Coordinate system for a 2D type loosely coupled mechanism

calculated using the expression

$$\theta = Ad^2 + Bd + C, \quad (2)$$

in which where the coefficients A, B and C are calibrated as 0.349, -5.525 and 0.753, respectively.

To prevent any slips, the top surface of the cartilage adheres to the rigid link. Under these conditions, the range of movement was about ± 20 deg.

B. Control law

A coordinate system was developed for a 2D type loosely coupled mechanism. The angle θ of the link was regulated using a simple P control law for the link (Figure 15):

$$v_{\text{inp}}^1 = \begin{cases} -K_P(\theta(t) - \theta_d) + v_{\text{offset}} & \text{when } \theta(t) < \theta_d \\ 0 & \text{when } \theta(t) \geq \theta_d \end{cases},$$

$$v_{\text{inp}}^2 = \begin{cases} 0 & \text{when } \theta(t) < \theta_d \\ -K_P(\theta(t) - \theta_d) + v_{\text{offset}} & \text{when } \theta(t) \geq \theta_d \end{cases}.$$

where θ_d is the desired angle, to be held constant, K_P is the proportional gain, which was always a positive constant, and v_{offset} was the minimum voltage required for the drivers, 1.7 V for our drivers. The load on the actuator complicates the relationship between input voltage and the actuator's driving force. Therefore, we used the input voltage to the driver IC as the control input. Since SMA actuators only contract during heating, but cannot expand autonomously, this control law includes conditional sentences. An SMA actuator with no voltage works as a bias spring in this mechanism.

Viscoelastic objects have been found to cause the center of rotation of the link to shift. This can lead to the control of two degrees of freedom, the angle and position of the link for the socket using two actuators. Thus, this mechanism has a redundancy in the control law. We did not control the shift of the link for the socket. The link position converges to a local minimum condition of elastic force in the cartilaginous area [14].

In addition, proportion laws do not depend on knowledge of the physical properties of the joint mechanism because, in the case of soft materials, the physical properties, especially the viscosity, are not accurately known. We did not apply velocity of the joint because the mechanism has natural damping components [15].

C. Experimental results

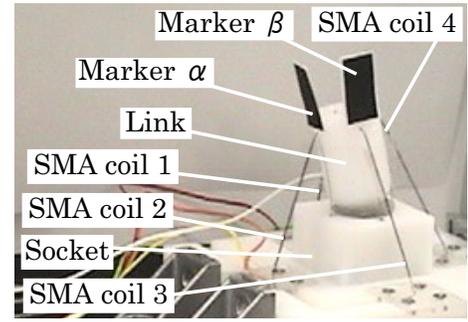
Figure 16 shows typical experimental results, in which the desired angle θ_d was set at 0, ± 5 , and ± 15 deg. To confirm the measurement precision of the length sensor, we compared these results with angles obtained by CCD images, with the latter calculated from the momentum of the second order of the rectangle black marker. Although each link converged accurately when the desired angle was negative, the angles calculated by CCD images and length sensors differed when the desired angles were positive, indicating that measurement errors occur when the desired angles are positive. Jamming can occur when the wire is pushed into the tube, which may cause estimation errors during positive rotation of the mechanism. The simplest way to decrease measurement errors is to utilize pairs of sensors that work antagonistically. We have utilized an arrangement of antagonistic sensors in a prototype of a 3D loosely coupled mechanism.

VI. ANGLE CONTROL USING LENGTH SENSORS IN 3D SPACE

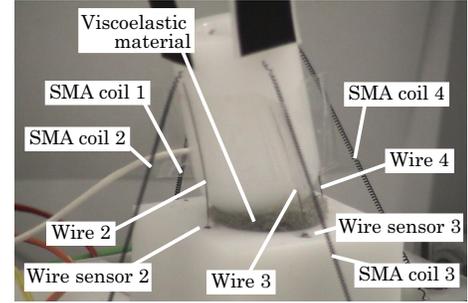
A. Angle measurement

Based on the previous 2D joint mechanism, we demonstrate angle control of a 3D joint mechanism using length information. Wire pulse coders were applied as in the 2D joint mechanism, along with two antagonistic pairs of SMA actuators (four SMA coils) arranged orthogonally (Figure 17). Figure 18 shows a coordinate system for the 3D type loosely coupled mechanism.

We regulated the projecting angles α and β of the link for the two reference projection planes, α side and β side, respectively. The sensors were arranged against each side of the socket, with SMA coils 1 and 3 and length sensors 1 and 3 were applied on the α side and SMA coils 2 and 4 and length sensors 2 and 4 on the β side (Figure 19). In this mechanism, both the actuators and sensors had redundancies for controlling the projecting angles. Although the link can be represented in roll-pitch-yaw notation, the relationship between the link angles and measurements obtained by the sensors is complicated, since the mechanism has a deformable cartilaginous area. If projection angles α and β can be controlled independently without any interferences, the projecting angles could be controlled simply and viscally.



(a) Actuator configurations



(b) Sensor configurations

Fig. 17. Loosely coupled mechanism of 3D type with length sensors

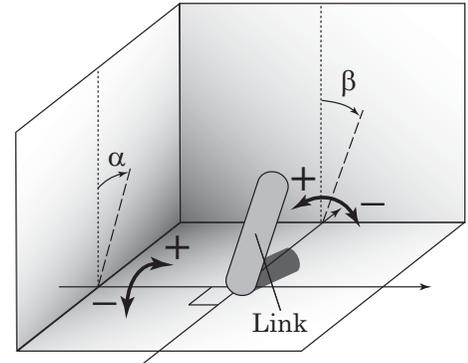


Fig. 18. Coordinate system for 3D type loosely coupled mechanism

Although the viscoelastic object causes a shift in the center of rotation of the link to shift, we did not control the shifting and twisting of the link against the socket.

To determine the relationship between the link angle and the length sensor, we calibrated the projecting link angles using CCD images. Figure 20 shows the angle identifications for the length sensors. Using an insert distance of the i -th sensor wire d_i , link angles α and β could be calculated as:

$$\begin{aligned}\alpha &= A(d_3 - d_1) + B, \\ \beta &= C(d_4 - d_2) + D,\end{aligned}\quad (3)$$

where the coefficients A , B , C and D were 2.727, 0.323, 2.670 and 0.131, respectively.

For simplicity, the top surface of the cartilage adhered to the rigid link. Under these conditions, the ranges of movement against each side were about ± 20 deg.

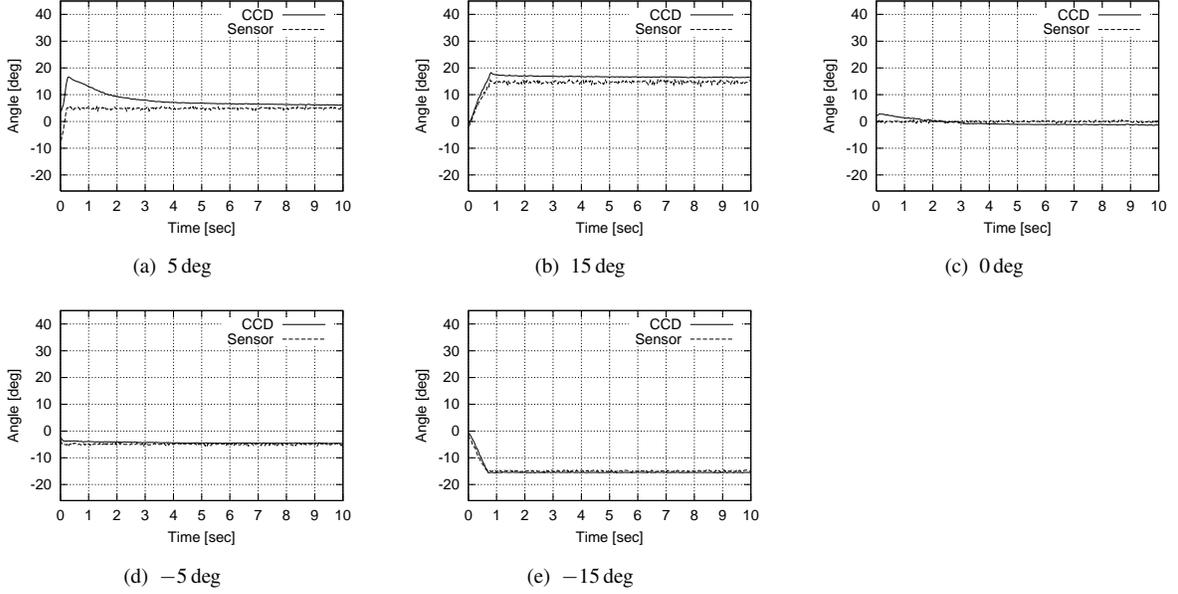


Fig. 16. Experimental results of a 2D mechanism using length feedback

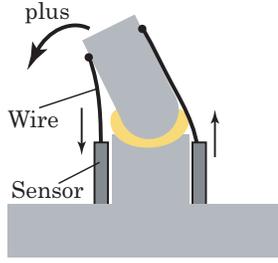


Fig. 19. Sensor configuration for 3D type loosely coupled mechanism

whereas the projecting angle β was driven by the simple P control laws:

$$v_{\text{inp}}^2 = \begin{cases} -K_{P2}(\beta(t) - \beta_d) + v_{\text{offset}} & \text{when } \beta(t) < \beta_d \\ 0 & \text{when } \beta(t) \geq \beta_d \end{cases},$$

$$v_{\text{inp}}^4 = \begin{cases} 0 & \text{when } \beta(t) < \beta_d \\ -K_{P4}(\beta(t) - \beta_d) + v_{\text{offset}} & \text{when } \beta(t) \geq \beta_d \end{cases},$$

where K_{P_i} is a proportional gain for the i -th SMA actuator, and α_d and β_d are desired projecting angles. The projecting angles $\alpha(t)$ and $\beta(t)$ were calculated from Eq. 3. Although the mechanism is a multi input-output system, we applied a set of single input-output as control laws. In general, viscoelastic objects have nonlinear physical parameters; hence, the identification of the parameters is difficult. If a precise model is constructed, the control laws tend to be complicated. Based on this scheme, we applied a single input-output set as a control law; using this set without physical parameters of the system, we could easily select feedback gains [?].

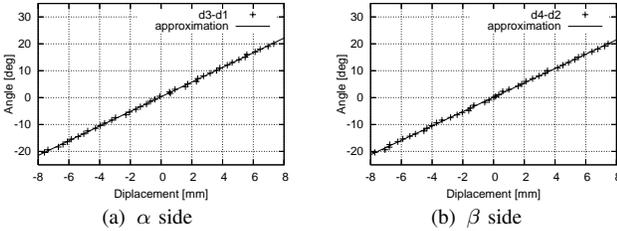


Fig. 20. Relationships between wire sensors and link angles

B. Control law

The projecting angle α was driven by the simple P control laws:

$$v_{\text{inp}}^1 = \begin{cases} -K_{P1}(\alpha(t) - \alpha_d) + v_{\text{offset}} & \text{when } \alpha(t) < \alpha_d \\ 0 & \text{when } \alpha(t) \geq \alpha_d \end{cases},$$

$$v_{\text{inp}}^3 = \begin{cases} 0 & \text{when } \alpha(t) < \alpha_d \\ -K_{P3}(\alpha(t) - \alpha_d) + v_{\text{offset}} & \text{when } \alpha(t) \geq \alpha_d \end{cases}.$$

C. Experimental results

Figure 21 shows typical experimental results. In Figure 21 (a) and (b), the desired angles α_d and β_d were set to 0 deg and 0 deg, respectively; in Figure 21 (c) and (d), the desired angles α_d and β_d were set to 5 deg and 5 deg, respectively; and in Figure 21 (e) and (f), the desired angles α_d and β_d were set to 10 deg and 10 deg, respectively. We compared the projecting angles obtained from CCD images to confirm the measurement precision of length sensors in these figures. The angles obtained from CCD images were calculated from the momentum of the second order of the rectangle black marker for each projecting plane. Positioning precision at the origin improved in comparison to using one length sensor in Figure 21 (a), (b), and Figure16 (c). We also found that the

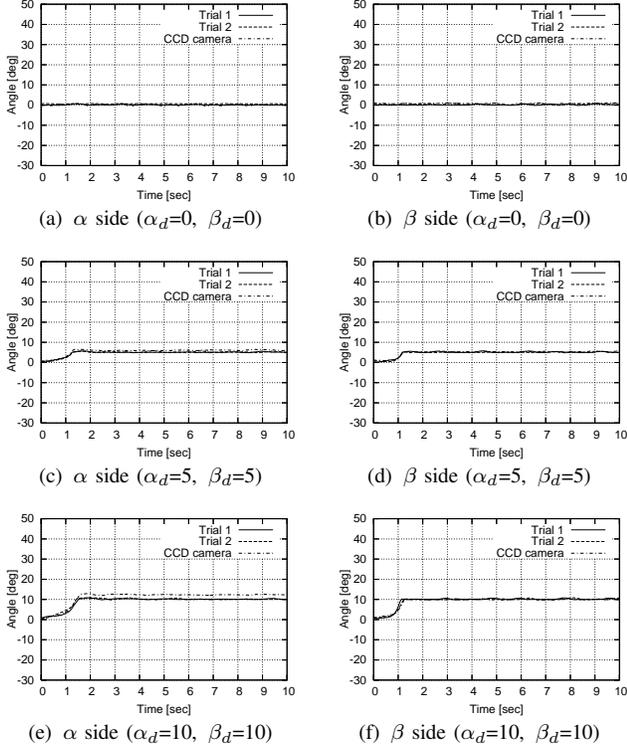


Fig. 21. Experimental results of a 3D mechanism using length feedback

3D loosely coupled mechanism tended to have measurement errors between the angle calculated by the length sensors and the angle measured using a CCD image when the desired angles were large. There may be interference between two pairs of antagonistic SMA actuators, or a wire pushing into a sensor tube may become jammed.

D. Improvement of angle measurement methods

We found that the 3D loosely coupled mechanism tended to show differences between the angle measured using length sensors and the angle obtained from a CCD image when the desired angles are set at large values. To reduce errors in measured angles we have implemented a scheme to select appropriate measurement methods. For each projecting plane on the α and β sides, we measured the values only on the right side of the length sensor, and those only on the left side of the length sensor, and determined the differences between the two length sensors. Figure 22 shows comparisons of the measurement angle errors using these measurement methods. Each curve in the figure could be approximated using a quadratic polynomial. When we applied only one length sensor, the measurement errors were large due to wire sensor jamming while pushing into the sensor tube. Based on these figures, we select the appropriate method to minimize angle errors. From Figure 22 (a), the measurement errors of angle $\alpha(t)$ could be reduced using the equations:

$$\begin{cases} \alpha(t) = A_0(d_3 - d_1) + B_0 : & \text{when } \alpha(t) \geq \alpha_{th} \\ \alpha(t) = A_1d_3^2 + B_1d_3 + C_1 : & \text{when } \alpha(t) < \alpha_{th} \end{cases}, \quad (4)$$

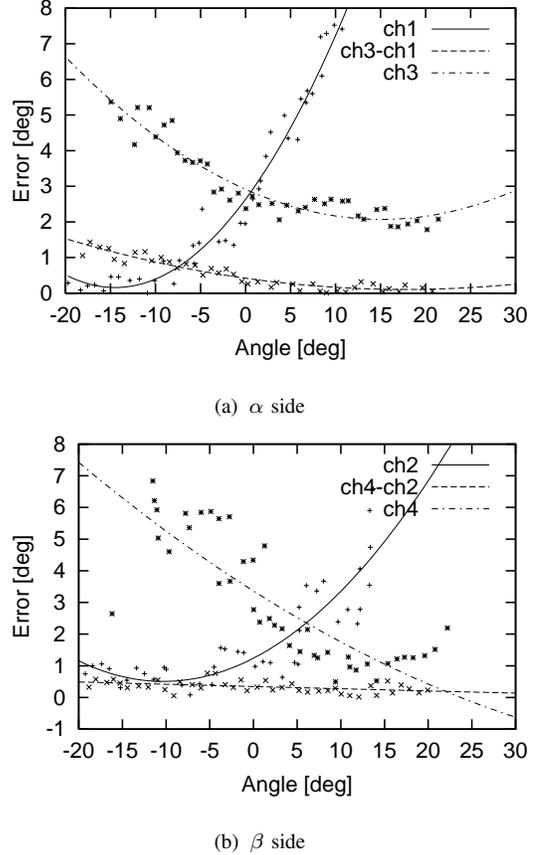


Fig. 22. Relationship between measurement error and method

where α_{th} is the threshold value for the angle, and A_j , B_j , and C_j ($j = 0, 1$) are constant values obtained by angle calibrations. In this case, the threshold α_{th} was -7.46 deg.

From Figure 22 (b), the measurement errors of angle $\beta(t)$ could be reduced using the equation:

$$\beta(t) = A_2(d_3 - d_1) + B_2, \quad (5)$$

where A_2 and B_2 are constant values obtained by angle calibrations. When this estimation method was applied, the measurement errors were not dependent on individual differences of the mechanism. For each projecting plane, the measurement angle errors were less than 1.0 deg in our 3D prototype.

VII. SUMMARY

We have proposed a novel link mechanism to investigate effects of soft component in control of a human-like joint. This joint mechanism, which uses a viscoelastic object and soft actuators instead of the cartilage and muscles of the human arm, is referred to as a loosely coupled mechanism. We confirmed that the viscoelastic object causes a shift in the center of rotation of the link, allowing the link to move smoothly, similar to a human joint. In addition, we used visual feedback to realize an angle control of the link mechanism. Because the joint has high compliance and damping characteristic under proportional (P) control, the

motion of the joint was highly robust against disturbances. We also showed that a 3D loosely coupled mechanism could be controlled using length sensors. We first constructed a prototype of the mechanism in 2D space to confirm motion of the link using one length sensor. Based on the prototype of the 2D mechanism, we constructed a 3D loosely coupled mechanism with length sensors. Using the 3D prototype, we were able to control two projecting angles. Finally, we implemented a scheme to select an appropriate measurement method to reduce errors measured by the length sensors. For each projecting plane, the errors were less than 1.0 deg in our 3D prototype.

Future work should include the construction of a robotic hand with loosely coupled mechanisms applying length sensors. This construction will use vast actuators instead of the muscle bundles of a human arm, both of which yield a large generative force. We also intend to verify the effects of slips between the rigid link and the cartilage. In this paper, in order to prevent any slips, the top surface of the cartilage adhered to the rigid link. In real human arms, the upper limb is not adherent to the cartilage, and arm motion includes smooth slips. In addition to constructing a prototype taking this property into account, we also intend to construct a dynamic model and to analyze the stability of the control laws. Future studies will provide the necessary guidelines for selecting an appropriate material and mechanism for the cartilaginous area of the link mechanism.

REFERENCES

- [1] M. Okada, Y. Nakamura and S. Hoshino: "Development of the Cybernetic Shoulder –A Three DOF Mechanism that Imitates Biological Shoulder-Motion–", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 543-548, 1999.
- [2] M. Okada, Y. Nakamura and S. Hoshino: "Design of Active/Passive Hybrid Compliance in the Frequency Domain", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 2250-2257, 2000.
- [3] Y. C. Fung: "Biomechanics: Mechanical Properties of Living Tissues", Springer-Verlag, 1993.
- [4] R. Hopler and M. Thummel: "Symbolic Computation of the Inverse Dynamics of Elastic Joint Robots", Proc. 2004 IEEE Int. Conf. on Robotics and Automation, pp.4314-4319, 2004.
- [5] C. J. B. Macnab, G. M. T. D'Eleuterio: "Discrete-Time Lyapunov Design for Neuroadaptive Control of Elastic-Joint Robots", The International Journal of Robotics Research, vol.19, No.5, pp.511-525, 2000.
- [6] M.W. Spong: "Modeling and control of elastic joint robots," ASME J. of Dynamic Systems, Measurement, and Control, vol. 109, no. 4, pp. 310-319, 1987.
- [7] A. Luca, R. Farina, and P. Lucibello: "On the Control of Robots with Visco-Elastic Joints", Proc. 2005 IEEE Int. Conf. on Robotics and Automation, pp. 4297-4302, 2005.
- [8] I. Mizuuchi, R. Tajima, T. Yoshikai, D. Sato, K. Nagashima, M. Inaba, Y. Kuniyoshi and H. Inoue: "The Design and Control of the Flexible Spine of a Fully Tendon-Driven Humanoid Kenta", Proc. of IEEE Int. Conf. on Intelligence Robots and Systems, pp. 2527-2532, 2002.
- [9] I. Mizuuchi, S. Yoshida, M. Inaba and H. Inoue: "The development and control of a flexible-spine for a human-form robot", Advanced Robotics Vol.17, No.2, pp179-196, 2003.
- [10] A. Vaz, and S. Hirai: "A Simplified Model for a Biomechanical Joint with Soft Cartilage", Proc. 2004 IEEE Int. Conf. on Systems, Man, and Cybernetics, 2004.
- [11] K. Shimizu and S. Hirai: "CMOS+FPGA Vision System for Visual Feedback of Mechanical Systems", Proc. IEEE Int. Conf. on Robotics and Automation, pp.2060-2065, 2006.
- [12] P.V. Komi, and C. Nicol: "Stretch-Shortening Cycle Fatigue", Biomechanics and biology of movement, pp.385-408, Human Kinetics, 2000.
- [13] K. Ikuta, M. Tsukamoto and S. Hirose: "Shape Memory Alloy Servo Actuator System with Electric Resistance Feedback and Application for Active Endoscope", Proc. of the 1988 IEEE Int. Conf. on Robotics and Automation, pp.427-430, 1988.
- [14] T. Inoue and S. Hirai: "Study on Hemispherical Soft-Fingered Handling for Fine Manipulation by Minimum D.O.F. Robotic Hand", Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation, pp. 2454-2459, 2006.
- [15] M. Shibata, T. Yoshimura and S. Hirai: "Loosely Coupled Joint Driven by SMA Coil Actuators", Proc. of IEEE Int. Conf. on Robotics and Automation, pp.4460-4465, 2007.