2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Interactive Motion Control of Deformable Objects Using Localized Optimal Control

Hongjun Jeon and Min-Hyung Choi

***Abstract*—In this paper we present a novel interactive method and interface techniques for controlling the behavior of physically-based simulation of deformable objects. The goal of our research is to provide users an ability to control the motion which appears physically correct, preserves the moving pattern of the original motion, and satisfies goals for a deformable object. In our approach, a user can select any part of the deformable structure, called control points, and can define target poses by moving control points. A user also can define target poses then our system automatically generates the motion path to achieve the target pose. With this technique patient specific organ simulation can be achieved by using a stream of image data. A series of sectional images can be the target poses. The optimal path generator computes the required control parameters that steer the intended node to the desired goal position while preserving the moving pattern of the original motion. It guarantees that the edited motion is physically conforming and natural.**

I. Introduction

Physically-based simulation techniques have been widely used in dynamic simulation of robots. While most of existing dynamic simulation of robots has focused on either rigid bodies or articulated structures, deformable object simulation is becoming increasingly important in industrial and surgical applications [1, 8, 28]. One way to approximate flexibility is to approximate the robot as an articulated robot, a linkage of rigid bodies. While this representation is sufficient for some situations, a more flexible representation is needed for situations that require robots to have larger deformations. Ability to control the behavior of deformable objects is a very important feature since it can provide a framework where we can try many "what if" questions and users can steer the simulation based on the intended target poses.

However, the direct and precise control of the behavior and trajectories of objects is difficult because it is based on a passive simulation model where the initial parameters and external forces are set upfront to compute the future motion. Motion of deformable objects is mostly generated using passive dynamic simulation because manually generating natural-looking deformation is extremely difficulty and time consuming. On the other hand, it is also difficult to control the trajectory and the final resting location of a simulated deformation object. Under the current passive dynamic simulation paradigm, a small adjustment of the initial parameters can drastically affect the subsequent motion and result in different behaviors in the end.

Another important cause of the difficulties in predicting and controlling the behavior of a deformable structure is that the widely used physical modeling techniques involve too much approximation and simplification to enhance the efficiency and programmability. The outcome of a simulation, using both the mass spring model and the finite element method (FEM), is heavily dependent to the combination of an element structure, the topology of connectivity, the granularity of elements, and coefficients that define the mechanical properties of the target deformable objects. If the granularity of the element structure is fine enough and all involved physical phenomena are precisely modeled, more realistic simulation can be achieved, but still it is insufficient to guarantee an exactly matched motion. Therefore, due to the limitation of the simplified modeling methods, certain behavior is somewhat unobtainable unless with a long, tedious trials and errors of parameter tweaking. As long as we use approximated models like mass-spring or FEM, localized motion editing is inevitable to get the exactly intended behavior and therefore an intuitive, interactive controllable simulation is important and desirable.

Controlling the dynamic behavior to a user's liking is very challenging since it requires sophisticated algorithms that steer the simulation by managing space and time dependent control forces over the neighboring structures on the fly. The thrust of this work is to automatically generate a set of appropriate control forces from user interaction metaphors. We adopt the two-phase simulation model: a pilot first-phase simulation with tentative coefficients is conducted and let the user edit the behavior of target nodes directly so that a desired behavior can be achieved in the second-phase simulation with added control parameters. Instead of repeating the initial parameter tweaking, we propose to use a set of time-encoded localized control parameters to steer the simulation. The proposed path generation algorithm automatically computes the motion paths and the required control parameters for the designated node and surrounding affected nodes, so that the goals can be satisfied in the second phase.

An attractive feature of our system is that the user is able to directly edit the motion at any time without adjusting the underlying physical parameters. The effect of control and behavioral changes are localized to the intended time and space without altering the material properties or fundamental conditions of the simulation. The proposed method generates a path for the deformable object by applying control forces to

Hongjun Jeon is with the Electrical and Computer Engineering Department, University of Colorado at Boulder, CO 80309 USA (corresponding author to provide phone: 303-556-8119; e-mail: jeon@ colorado.edu).

Min-Hyung Choi is with the Computer Science and Engineering Department, University of Colorado at Denver and Health Sciences Center, CO 80217 USA (e-mail: Min.Choi@cudenver.edu).

the deformable object until it reaches a configuration near the goal configuration. The main contribution of our locally optimal method is the substantial performance improvement and ease of implementation for the path generation by removing the expensive gradient computation of conventional optimal methods. Typically the gradient computation has been the bottleneck for computing control parameters because each parameter required its own derivative computation at every time step.

II. Related Work

Motion sketching and relevant interactive technique for controlling and manipulating rigid body simulation are introduced in [25, 26]. To achieve a desired motion, they changed initial physical parameters (position, velocity, etc.), so the result of editing affects the simulation globally due to the tweaked fundamental parameters for the entire simulation. This method works well for rigid bodies but it is difficult to apply the same technique to deformable objects because of the large number of degrees of freedom and associated computational complexity in a deformable object. Motion synthesis techniques that generate motions by cutting and pasting motion capture data [2, 3], parameterization of motions [19] and path-based editing of existing motion data [15] were successfully applied to human motion synthesis. Similarly an inverse kinematics system, based on a learned model of human poses, was introduced in [16] but it is also limited to articulated figures like human body. James et al. [17] used pre-computation based data driven tabulation of state space for quick computation of shape and appearance of deformable objects. Their method permits real-time hardware synthesis of nonlinear deformation and real-time user interaction. But the pre-computing process to encompass possible future motions requires a heavy computation and it only covers a small portion of frequently animated modes. Constrained dynamic schemes have been widely studied in robotics and computer graphics for motion control. Putting a proper set of geometric constraints over a dynamic system is often used to control the behaviors of dynamic system [4, 5, 9, 14, 22]. Constraint-based methods work well for enforcing geometric constraints but it is very difficult to generate particular motions since the desired motion is in a higher abstract form and it should be parameterized with a set of proper constraints. Realistic simulation and motion control of smoke and water has drawn a lot of attention from the special effects industry. These phenomena require high-level control mechanisms for physics-based fluid simulations; for example, keyframe fluid simulation [21, 27], target-driven simulation [11], and level-set method [10, 12, 20, 23]. While these studies were successful in achieving the desired configuration of fluid, still the generation of optimal paths to form the configuration and an interactive editing in the middle of a simulation remain a significant challenge.

III. Motion Control

In most motion control task, the first problem is to reach the desired position. To meet the goal, the control vector $\boldsymbol{u}$ = $[\boldsymbol{u}_0\ \boldsymbol{u}_1\ \dots\ \boldsymbol{u}_{N\text{-}1}]$ that guarantees meeting the goal positions within a finite time frames must be calculated. There are N unknown control forces but we only have one given condition (The final position, $\boldsymbol{q}_N$, is equal to the desired position, $\boldsymbol{q}_N^{\#}$). It is not enough condition to solve the problem and infinite number of solutions are possible so additional conditions are required. We create the cost function $\boldsymbol{J}$ which measures required energy and evaluates the goal satisfaction criteria. In addition the time range N over which the control forces are engaged should be defined. In our simulation, the first simulation (the original motion) is run with tentative coefficients then a user can select the motion at any time and edit the motion by moving selected nodes (control points). The target poses also can be predefined or patient specific sectional images. The target motions can be achieved in the second simulation with added control forces. Another condition is that the new path should preserve the moving pattern of the original motion. If the objective is to minimize the energy only, then it may violate the moving pattern of the original motion. It may generate a jerky motion with sharp turns which cause discontinuity and uneven velocity distribution, resulting unnatural motion.

A. The Deformable Model and Numerical Integration

Mass-spring, FEM and point-based systems are among the most commonly used strategies for building deformable objects. Mass-spring systems that we have chosen to work with are one of the most common forms for modeling deformable objects due to its simplicity and efficiency. The basic concept of mass-spring system is the deformable object is discritized into set of mass points which are connected by springs and dampers. Structural springs are connected in a rectangular format and shear springs are connected in diagonal direction. Primary and secondary bending spring structures are similar to [7].

For each mass point, the governing equation is written as

$$m_i \ddot{\boldsymbol{q}}_i + \sum_j \boldsymbol{g}_{ij} = \boldsymbol{f}_i \qquad (1)$$

Here m is mass of a point node, $\boldsymbol{q}$ is position of a node, $\ddot{\boldsymbol{q}}$ is acceleration of a node, $\boldsymbol{g}$ is spring and damping force between springs and $\boldsymbol{f}$ is net forces. Equation (1) expresses Newton's second law of motion for discrete masses. The spring force $\boldsymbol{g}$ is calculated by

$$\boldsymbol{g}_{ij} = -\left[k_s\left(|\Delta\boldsymbol{q}| - r\right) + k_d\left(\frac{\Delta\boldsymbol{v}\cdot\Delta\boldsymbol{q}}{|\Delta\boldsymbol{q}|}\right)\right]\frac{\Delta\boldsymbol{q}}{|\Delta\boldsymbol{q}|} \qquad (2)$$

where k_s is spring stiffness constant, k_d is damping coefficient, $\Delta\boldsymbol{q} = \boldsymbol{q}_i - \boldsymbol{q}_j$, and $\Delta\boldsymbol{v} = \boldsymbol{v}_i - \boldsymbol{v}_j$ ($\boldsymbol{v}$ is velocity of a node). We use a second order implicit integration as our numerical integration scheme as described in [7] and shown as

$$\frac{1}{h}\begin{bmatrix} \frac{3}{2}\boldsymbol{q}_{n+1} - 2\boldsymbol{q}_n + \frac{1}{2}\boldsymbol{q}_{n-1} \\ \frac{3}{2}\boldsymbol{v}_{n+1} - 2\boldsymbol{v}_n + \frac{1}{2}\boldsymbol{v}_{n-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_{n+1} \\ \boldsymbol{M}^{-1}\boldsymbol{F}_{n+1} \end{bmatrix} \qquad (3)$$

where h is time step and $\boldsymbol{F}$ is the force vector. The nonlinear term $\boldsymbol{F}_{n+1}$ by using a first order Taylor series approximation is replaced with

$$\boldsymbol{F}_n + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}(\boldsymbol{q}_{n+1} - \boldsymbol{q}_n) + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}(\boldsymbol{v}_{n+1} - \boldsymbol{v}_n) \tag{4}$$

where $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}$ and $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}$ are the Jacobian matrices of the particle forces with respect to position and velocity. By combining equation (3) and (4), we can obtain a linear system. If we rearrange the linear system, the equation becomes,

$$\begin{aligned}\left(\mathbf{I} - \frac{2}{3}h\boldsymbol{M}^{-1}\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}} - \frac{4}{9}h^2\boldsymbol{M}^{-1}\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}\right)(\boldsymbol{q}_{n+1} - \boldsymbol{q}_n) \\ = \frac{1}{3}(\boldsymbol{q}_n - \boldsymbol{q}_{n-1}) + \frac{h}{9}(8\boldsymbol{v}_n - 2\boldsymbol{v}_{n-1}) \\ + \frac{4}{9}h^2\boldsymbol{M}^{-1}\left(\boldsymbol{F}_n - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}\boldsymbol{v}_n\right) - \frac{2}{9}h\boldsymbol{M}^{-1}\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}(\boldsymbol{q}_n - \boldsymbol{q}_{n-1})\end{aligned} \tag{5}$$

The second order implicit method is more stable and large time step can be used in our simulation.

B. Control Parameters

We define a control vector, $\boldsymbol{u}$, which encodes all the external influences the system has over simulation. The physical system equation of movement of discrete masses with external control forces is

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{F} + \boldsymbol{u} \tag{6}$$

where $\boldsymbol{F}$ are applied, gravitational, damping, and spring forces acting on the discrete masses. With the Euler method, the equation of motion (equation (6)) along with the kinematic relationship between $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ is discretized as

$$\dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}_n + \Delta t\boldsymbol{M}^{-1}\boldsymbol{F}_n + \Delta t\boldsymbol{M}^{-1}\boldsymbol{u}_n \tag{7}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \Delta t\dot{\boldsymbol{q}}_{n+1} \tag{8}$$

Substituting $\dot{\boldsymbol{q}}_{n+1}$ from equation (7) into equation (8) we obtain

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \Delta t\dot{\boldsymbol{q}}_n + \Delta t\left[\Delta t\boldsymbol{M}^{-1}\boldsymbol{F}_n + \Delta t\boldsymbol{M}^{-1}\boldsymbol{u}_n\right] \tag{9}$$

$$\boldsymbol{u}_n = \boldsymbol{M}\left\{\boldsymbol{q}_{n+1} - \left[\boldsymbol{q}_n + \Delta t\dot{\boldsymbol{q}}_n + \Delta t^2\boldsymbol{M}^{-1}\boldsymbol{F}_n\right]\right\}/\Delta t \tag{10}$$

Our system tries to generate the new motion path $\boldsymbol{q}^*$ using the locally optimal method, computed by the control forces described in equation (10). These control forces should guarantee the preservation of original motion pattern and meeting the goal poses.

C. Motion Path Generation using the Locally Optimal Control

The formulation of an optimal control problem requires a mathematical model of the process to be controlled, a statement of the physical constraints, and specification of a performance criterion [6, 18]. Equation (6) is our mathematical model. The next step is to define the physical constraints on the state and control values but the time range N which a user wants to control must be defined in advance. A user selects one node which a user wants to move at current time T_c and simply drag it to a desired location $\boldsymbol{q}_N^\#$. We set a time range to be directly proportional to the length of change between the current position $\boldsymbol{q}_N$ and the desired position $\boldsymbol{q}_N^\#$ (See Figure 1).

$$N = \alpha\left\|\boldsymbol{q}_N^\# - \boldsymbol{q}_N\right\| \tag{11}$$

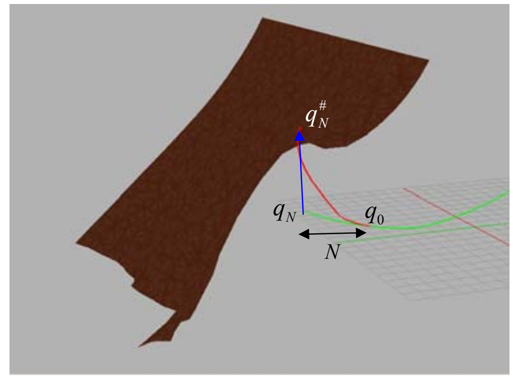


Fig. 1. This figure shows how to set the time range N. We set a time range to be directly proportional to the length (blue line) of change between the current position and the desired position.

where α is a scale factor. The physical constraints are that the final state of a system, $\boldsymbol{q}_N^*$, must be in a specified regions $\boldsymbol{D}$ called target set and the control forces must be less than and equal to the maximum control force F_{max}. When the control forces are included in the system to influence the simulation, there should be criteria to measure how well the user's desired path is met. The cost function $\boldsymbol{J}$ measures how closely the affected node reaches the desired position and also penalizes the system for using too much control forces. These physical conditions and the cost function can be written as:

$$\begin{aligned}\boldsymbol{J} = \frac{1}{2}\sum_{i=0}^{N-1}\left[\beta\left\|\boldsymbol{q}_N^\# - \boldsymbol{q}_{i+1}^*\right\|^2 + \gamma\left\|\boldsymbol{u}_i\right\|^2\right] \\ \text{subject to: } \left\|\boldsymbol{q}_N^\# - \boldsymbol{q}_N^*\right\| \subseteq \boldsymbol{D} \text{ and } \left\|\boldsymbol{u}_i\right\| \le F_{max} \text{ for } 0 \le i < N\end{aligned} \tag{12}$$

where $\beta \ge 0$ and $\gamma \ge 0$ are weighting factors. Large γ puts more emphasis on the amount of control force while β concerns about the goal satisfaction. The optimal control problem is to find an admissible control force u which causes the system to follow an admissible trajectory $\boldsymbol{q}^*$.

Conventional methods to find the control vector that minimizes the cost function typically use the Riccati equation [24] or a set of Lagrange multipliers [13]. Since those methods are gradient based, we must not only evaluate $\boldsymbol{J}$, but also compute its gradient $d\boldsymbol{J}/d\boldsymbol{u}$. This gradient computation for deformable structure is very costly, as the matrix $d\boldsymbol{J}/d\boldsymbol{u}$ consists of an entire state sequence for each control. A major drawback of these gradient based methods is the computation of a large linear system given that inverting a general n x n matrix is $O(n^3)$. Deformable structure typically involves thousands of nodes so the Jacobian matrix of them can be very large. We cast the problem to a combinatorial discrete optimization to avoid the expensive computation. Instead of solving the problem with the conventional nonlinear optimization method, our algorithm tries to find local radial vectors only for the new control force direction that can improve the cost function at every iteration step independently. This multistage decision process to get a locally optimal path is shown in Figure 2. Let us denote the scalar ratio L between the length of original path and the length of desired path;

$$L = \delta\frac{\left\|\boldsymbol{q}_0 - \boldsymbol{q}_N^\#\right\|}{\left\|\boldsymbol{q}_0 - \boldsymbol{q}_N\right\|} \tag{13}$$

where δ is a scale factor. Also we define the radius vectors $\boldsymbol{R}_i$;

$$\boldsymbol{R}_i = L\left[\boldsymbol{q}_{i+1} - \boldsymbol{q}_i\right] \quad \text{for} \quad 0 \le i < N \tag{14}$$

Then the algorithm initially sets a seek angle range $\boldsymbol{\theta}$ and searches for a condition that reduces the cost function by altering the search angle at a discrete interval, from $-\boldsymbol{\theta}$ to $\boldsymbol{\theta}$. $\boldsymbol{q}_{i+1,\theta}$ is the new point after the rotation $\boldsymbol{R}_i$ for $\boldsymbol{\theta}$ degrees centered at $\boldsymbol{q}_i^*$.

$$\boldsymbol{q}_{i+1,\theta} \triangleq \boldsymbol{q}_i^* + \mathbf{Rot}(\boldsymbol{\theta}) \cdot \boldsymbol{R}_i \tag{15}$$

where $\mathbf{Rot}(\boldsymbol{\theta})$ is a rotation matrix. Control forces $\boldsymbol{u}_{i,\theta}$ for every point $\boldsymbol{q}_{i+1,\theta}$ on the circle with the radius $\boldsymbol{R}_i$ can be calculated by equation (10) for $0 \le i < N$.

$$\boldsymbol{u}_{i,\theta} = \boldsymbol{M}\left\{\boldsymbol{q}_{i+1,\theta} - \left[\boldsymbol{q}_i^* + \Delta t \dot{\boldsymbol{q}}_i^* + \Delta t^2 \boldsymbol{M}^{-1} \boldsymbol{F}_i\right]\right\} \Big/ \Delta t \tag{16}$$

By repeating this process until the seek angle reaches the predefined boundary, locally optimal control forces $\boldsymbol{u}$ to minimize the cost function can be found and the motion path $\boldsymbol{q}^*$ can be generated. In our method, the computation of a large linear system is eliminated. The computation of our method mostly depends on the number of discretized seek angles.

$$\boldsymbol{min}(\boldsymbol{J}_{i,\theta}) = \boldsymbol{min}\left(\frac{1}{2}\left[\beta\left\|\boldsymbol{q}_{i+1,\theta} - \boldsymbol{q}_n^\#\right\|^2 + \gamma\left\|\boldsymbol{u}_{i,\theta}\right\|^2\right]\right) \tag{17}$$

The solutions of equation (17) are our newly obtained motion path $\boldsymbol{q}^*$. If the final state $\boldsymbol{q}^*$ does not reach to the target region $\boldsymbol{D}$, $\boldsymbol{\delta}$ must be further adjusted to reach to $\boldsymbol{D}$. Another goal that must be satisfied concurrently is to generate the path that reaches the desired position while preserving the moving pattern of the current motion. Finding the locally optimal control force, obtained in the radial vector form within the seek circle, is particularly convenient for preserving the moving pattern of the original motion since the arc length of new step is proportional to the step length of the original motion. Since the moving pattern of a motion is mostly dictated by the magnitude and direction of each step, radial vectors with proportionally scaled arc lengths, combined with minimally deviated seek angles, can help to conserve the moving pattern of the original motion in more straightforward manner.

IV. Multiple Paths Control and Spatial Distribution of Control Forces

A deformable object involves many particles and in order to control the behavior of the object, all affected particles need to be handled at the same time. Controlling multiple individual particles may incur conflicting control forces, or they may affect each other in unexpected ways since they are all connected in the mesh structure. To address the conflicting control forces issue we use a time-stamped sequential control force application for the overlapped control range of multiple paths control. For instance, when there are two particles' paths to be edited, the time range of the optimal path generation is redefined to deal with the possible interplay. Let's assume the time range of the first particle A is [t_1, t_2] and [t_3, t_4] for the particle B. If t_3 is earlier than t_1 the particle B's newly edited behavior will affect the particle A's force computation at time t_3. To avoid this conflict, t_1 is redefined to t_3 so that the two particles' mutual influences can be considered in the optimal paths generation. Figure 3 shows two optimally generated paths with overlapped time range.

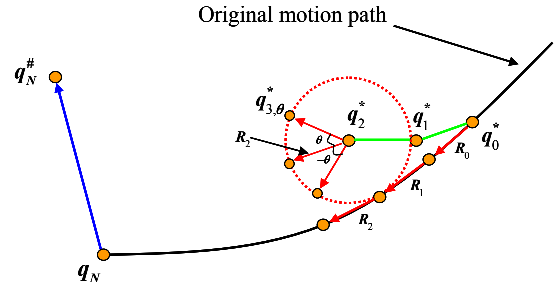


Fig. 2. This figure shows the optimal path generation process using the multistage decision process. The red circle is the region to find the locally minimum optimal position. $\boldsymbol{R}_i$ is the radius vector that determines the direction of control force. Green line is the newly generated locally optimal trajectory.

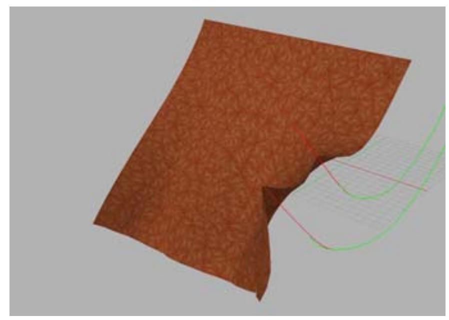


Fig. 3. This figure shows that the multiple goals are simultaneously satisfied and it displays two optimally generated paths. Green lines are original motion path and red lines are generated paths.

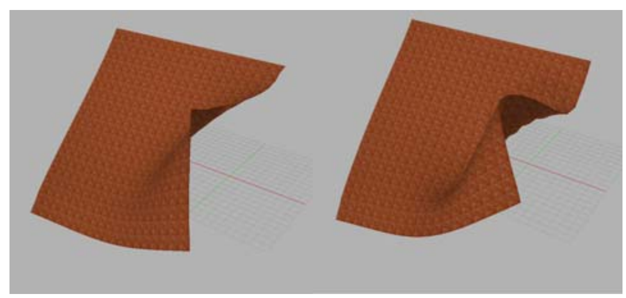


Fig. 4. This figure shows spatial distribution of control forces. Left: the control forces are applied to the selected node. Right: The control forces are spatially distributed to neighbor nodes

The examples in Figure 3 and figure 4 use a mesh with 400 nodes to illustrate a node based control. It is the screen capture of a live simulation of free falling soft thin shells. In those examples, a user can select any part of the deformable structure, move it to the desired position, and the system generates the path using our algorithm. Generated control forces are only applied to the selected node. The selected node and neighbor nodes are connected by springs. Neighbor nodes just follow the selected node. These generate very sharp final pose (figure4, left). To eliminate this problem we spatially distribute control forces to the neighbor nodes. Figure 4 shows the result of distributing control force to the neighbor nodes.

V. Experimental Results

We have performed skirt simulation with human walking motion in Figure 5. The example in Figure 5 uses a skirt with 1980 nodes to illustrate overall procedures of motion control. In this example, a user can select control points (green points in Figure 5 (b)) to be controlled and predefined curling motion that becomes the target pose (blue line in Figure 5 (b)), then out system generates the target motion using our algorithm (Figure 5 (c)). We also have applied the proposed controllable simulation to a patient specific heart model based on a generic heart motion. The human heart model has 3372 nodes. Figure 6 (a) shows a snapshot of a generic heart motion, (b) shows a patient specific sectional image and (c) shows a modified motion when a part of the heart was guided by patient specific geometric changes. It is a dynamic data driven simulation of patient specific organ simulation where the control terms are automatically extracted from a patient sectional image. This sectional image is a guided control input. We have another example of human heart simulation. In Figure 7, a user can select any node of the heart model to be controlled and define the target pose by moving control points (Figure 7 (b)). The final controlled motion is shown in Figure 7 (c). Those all examples are the screen capture of a live simulation.

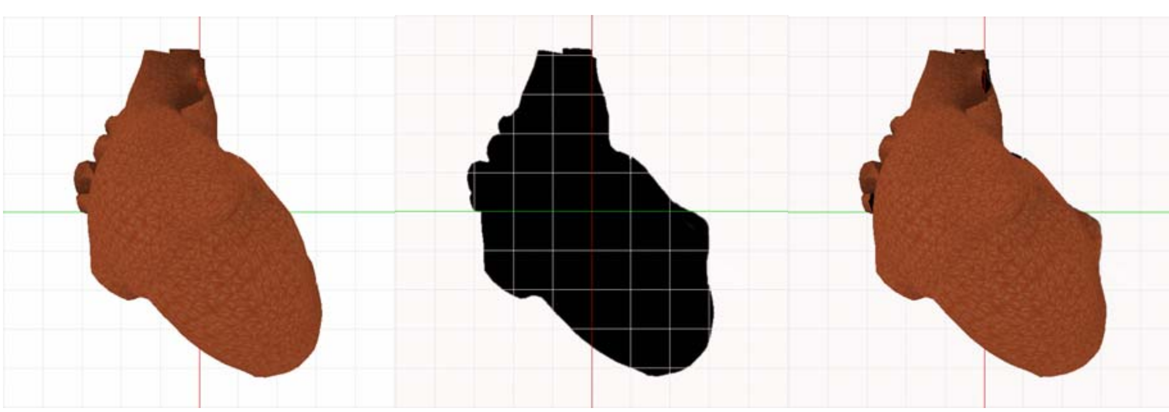
Even if the motion can be generated automatically, the process involves several user-defined coefficients such as β for the evaluation of the goal satisfaction, and γ for the admissible amount of control force. β and γ are dependent on the magnitude of distance error and control force. If γ is too large, that signifies small control forces applied to the system, the final position may not reach to the desired position. If β exceeds a threshold, the generated path tends to be the shortest path, and the route could be unnaturally straight. To avoid this, we limit the seek angle θ within minimal range at initial steps and gradually increase the range. Since appropriate coefficients are problem dependent, they can be adjusted by the system designer to reflect the intended motion.

VI. Conclusion and Discussion

This paper reports a locally optimal method to control the behavior of physically-based simulation of deformable objects. We have shown that the motion path and the required control parameters can be automatically generated and the new simulation in the second phase satisfies all user defined goals by the control parameters. In our simulation, the target poses can be defined by selecting and moving control points. In addition, a user can predefine target poses such as a curl motion. A series of sectional images can be the target poses so that patient specific simulation can be achieved. The optimal path generator computes the required control parameters that steer the intended node to the desired goal position. It helps us in preserving the moving pattern of the original simulation in more straightforward manner.

(a) (b) (c)

Fig. 5. These series of figures show how to control the behavior of a skirt simulation with human walking motion. The target pose (blue line in (b)) is predefined. (a) is a snapshot of the original motion. In figure (b), the red dots are nodes of the skirt model, green dots are control points which are selected by a user and the blue line is the predefined curl pose which is the target pose. (c) shows the final controlled motion generated by our system.



(a) (b) (c)

Fig. 6. These figures show a patient specific heart simulation. (a) shows a snapshot of a generic heart motion. (b) shows a patient specific sectional image. (c) shows a modified motion when a part of the heart was guided by patient specific geometric changes.

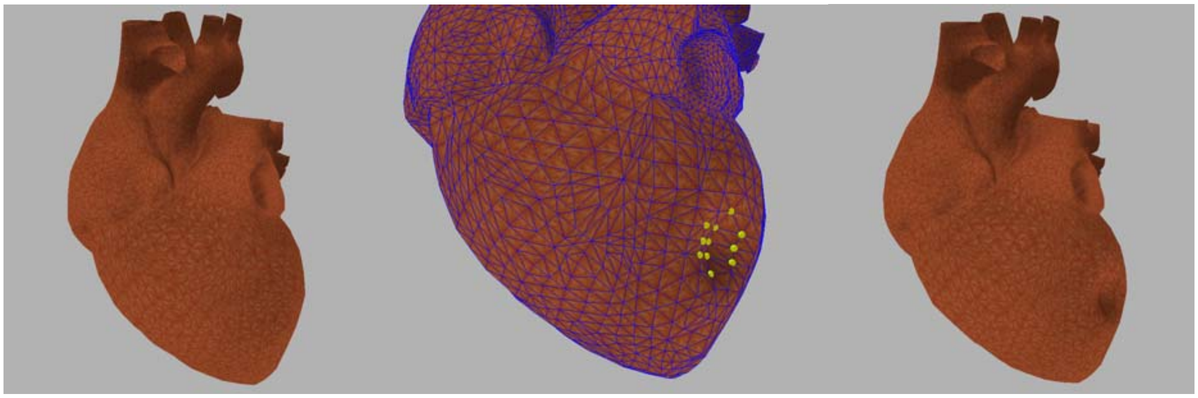


Fig. 7. Those figure show controllable simulation of a human heart motion. (a) shows a snapshot of a generic heart motion. In figure (b), the blue lines are the structure springs of human heart model and green dots are control points that are selected by a user. (b) shows the target pose which is defined by a user moving the control points. (c) shows the final controlled motion.

In the future, to optimize the search space and to accommodate the physical conditions of unreachable regions, new algorithms to detect the safe regions (physically possible regions) and a new scheme to incorporate this condition in the heuristic optimal path search needs to be further investigated. More studies on user interface to inform the user about the unreachable regions or accumulated collisions along the path is needed. If the desired motion is very complex or has large geometric changes, complex collision problems may be occurred. Collision detection for deformable objects is particularly difficult since the size and shape of the objects are continuously changing. We need to study how to handle those collision problems efficiently.

REFERENCES

[1] ALTEROVITZ, R., GOLDBERG, K., AND OKAMURA, A. 2005. "Planning for Steerable Bevel-tip Needle Insertion Through 2D Soft Tissue with Obstacles," *In Proc. of IEEE Int. Conf. Robotics and Automation*, 1652-1657.

[2] ARIKAN, O., AND FORSYTH, D. A. 2002. "Interactive motion generation from examples," *In Proc. of SIGGRAPH 2002,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 483-490.

[3] ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. F. 2003. "Motion synthesis from annotations," *In Proc. of SIGGRAPH 2003,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 402-408.

[4] BARAFF, D., AND WITKIN, A. 1998. "Large steps in cloth simulation," *In Proc. of SIGGRAPH 1998,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 1-12.

[5] BARZEL, R., AND BARR, A. H. 1988. "A modeling system based on dynamic constraints," *In Proc. of SIGGRAPH 1988,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 179-188.

[6] CHEN, C. T. 1999. *Linear system theory and design, Third edition*. Oxford Univ. Press.

[7] CHOI, K. J., AND KO, H. S. 2002. "Stable but responsive cloth," *In Proc. of SIGGRAPH 2002,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 604-611.

[8] CHOI, K. S., SUN, H., AND HENG, P. A. 2003. "Interactive Deformation of Soft Tissues with Haptic Feedback for Medical Learning," *In Proc. of IEEE Information Technology in Biomedicine*. Vol. 7, NO. 4, 358-363.

[9] CLINE, M. B. AND PAI, D. K. 2003. "Post-Stabilization for Rigid Body Simulation with Contact and Constraints," *In Proc. of IEEE Int. Conf. Robotics and Automation*, 1652-1657.

[10] ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surfaces. *In Proc. of SIGGRAPH 2002,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 736-744.

[11] FATTAL, R., AND LISCHINSKI, D. 2004. Target-driven smoke animation. *In Proc. of SIGGRAPH 2004,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 441-448.

[12] FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. *In Proc. of SIGGRAPH 2001,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 23-30.

[13] FRANKLIN, G. F., POWELL, J. D., AND WORKMAN, M. 1998. *Digital Control of Dynamic Systems.* Addison-Wesley.

[14] GAYLE, R., LIN, M. C., AND MANOCHA, D. 2005. Constraint-Based Motion Planning of Deformable Robots. *In Proc. of IEEE Int. Conf. Robotics and Automation*, 1058-1065.

[15] GLEICHER, M. 2001. "Motion path editing," *The 2001 ACM Symposium on Interactive 3D Graphics*.

[16] GROCHOW, K., MARTIN, S. L., POPOVIC, J., AND HERTZMANN, A. 2004. "Style-based inverse kinematics," *In Proc. of SIGGRAPH 2004,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 522-531.

[17] JAMES, D. L., AND FATAHALIAN, K. 2003. "Precomputing interactive dynamic deformable scenes," *In Proc. of SIGGRAPH 2003,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 879-887.

[18] JUANG, J. H. 1994. *Applied system identification.* Prentice-Hall, Inc.

[19] KOVAR, L., AND GLEICHER, M. 2004. "Automated extraction and parameterization of motions in large data sets," *In Proc. of SIGGRAPH 2004,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 559-568.

[20] LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. *In Proc. of SIGGRAPH 2004,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 457-462.

[21] MCNAMARA, A., TREUILLE, A., POPOVIC, Z., AND STAM, J. 2004. Fluid control using the adjoint method. *In Proc. of SIGGRAPH 2004,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 449-456.

[22] MIN, H., CHOI, M. H., JUNG, S., WELCH, S., AND TRAPP, J. 2005. "Effective Constrained Dynamic Simulation Using Implicit Constraint Enforcement," *In Proc. of IEEE Int. Conf. Robotics and Automation*, 4531-4536.

[23] NGUYEN, D., FEDKIW, R., AND JENSEN, H. 2002. Physically based modeling and animation of fire. *In Proc. of SIGGRAPH 2002,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 736-744.

[24] OGATA, K. 1997. *Modern Control Engineering, Third Edition.* Prentice-Hall.

[25] POPOVIC, J., SEITZ, S. M., ERDMANN, M., POPOVIC, Z., AND WITKIN A. 2000. "Interactive manipulation of rigid body simulations," *In Proc. of SIGGRAPH 2000,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 209-218.

[26] POPOVIC, J., SEITZ, S. M., AND ERDMANN, M. 2003. "Motion sketching for control of rigid body simulations," *ACM Transactions on Graphics*, Vol. 22, No. 4, 1034-1054.

[27] TREUILLE, A., MCNAMARA, A., POPOVIC, Z., AND STAM, J. 2003. Keyframe control of smoke simulations. *In Proc. of SIGGRAPH 2003,* ACM Press / ACM SIGGRAPH, Comput. Graphics Proc., 716-723.

[28] ZHONG, Y., SHIRINZADEH, B., ALICI, G., AND SMITH, J. 2005. "A New Methodology for Deformable Object Simulation," *In Proc. of IEEE Int. Conf. Robotics and Automation*, 1914-1919.