EUROGRAPHICS 2012 / P. Cignoni, T. Ertl
(Guest Editors)

Volume 31 (2012), Number 2

Manipulation of Flexible Objects by Geodesic Control

He Wang and Taku Komura

School of Informatics
The University of Edinburgh

Figure 1: *Snapshots of conducting a basket wrapping using Geodesic Control*

Abstract

We propose an effective and intuitive method for controlling flexible models such as ropes and cloth. Automating manipulation of such flexible objects is not an easy task due to the high dimensionality of the objects and the low dimensionality of the control. In order to cope with this problem, we introduce a method called Geodesic Control, which greatly helps to manipulate flexible objects. The core idea is to decrease the degrees of freedom of the flexible object by moving it along the geodesic line of the object that it is interacting with. By repeatedly applying this control, users can easily synthesize animations of twisting and knotting a piece of rope or wrapping a cloth around an object. We show examples of "furoshiki wrapping", in which an object is wrapped by a cloth by a series of maneuvers based on Geodesic Control. As our representation can abstract such maneuvers well, the procedure designed by a user can be re-applied for different combinations of cloth and an object. The method is applicable not only for computer animation but also for 3D computer games and virtual reality systems.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Simulation of flexible objects such as ropes and cloth is a rapidly growing area with many applications in computer games, movies and design. However, there have not been many solutions proposed for controlling these objects during maneuvers such as winding, knotting and wrapping. For example, a wrapping task called basket wrapping involves many steps as shown in Fig. 1: putting the object at the center of the cloth, bringing its two corners and knotting it under the handle of the basket, then winding the other two corners around the handle and knotting them at the top. Although humans can easily demonstrate and explain such processes to one another, automating such maneuvers and robustly controlling the cloth to reach the target configuration is not an easy task.

The main source of the problem is the high dimensionality of the flexible object. Traditionally, rope and cloth manipulations are guided through the 3D trajectories of the end points or the corners. The rest of the object is only passively affected by the movement of the controlled area. However, the number of degrees of freedom of a piece of cloth is very large. Controlling an object with a high dimensional state space using low dimensional control signals is difficult. In addition, the trajectories of the end-points must be planned by global path-planning approaches to make sure that the flexible object does not get hooked with obstacles or with itself.

In order to cope with this problem, we propose a method to abstract the control of the flexible object. More specifically, it is a control method called Geodesic Control that can significantly decrease the degrees of freedom of the flexible object while also ensuring the resultant movement is always predictable. This approach is based on the observation of the method that humans use to control flexible objects. For example, when winding one rope around another, generally people first cross them, and then gradually increase the contact area between the two ropes along the tangent surface of the ropes. This strategy is robust from the control point of view, as there is little redundancy at the region where the knot is produced. A similar idea can be used when wrapping a cloth around an object; the surface contact area can be gradually increased until the cloth covers the whole object. Using Geodesic Control, a significant reduction in the complexity of the control can be achieved. At the level of winding and wrapping, global path planning is not needed as the flexible object simply moves either along the geodecis line defined on the surface of the object or that of its convex hull.

Using Geodesic Control, the process of making "furoshiki wraps", a style of wrapping and knotting cloths to carry objects, can be abstracted to a simple Finite State Machine (FSM). The same FSM can be used to wrap objects of different shapes. The wrapping maneuvers can be synthesized in interactive time, which means the method is not only applicable for offline animation for films but also for interactive applications such as computer games and virtual reality systems.

Contributions

1. We propose a control approach called Geodesic Control, which significantly reduces the complexity and increases the robustness of winding and wrapping maneuvers.
2. We propose to abstract complex wrapping styles by a FSM composed of a series of Geodesic Control maneuvers. Using this FSM, we can automatically wrap objects of different geometries.

2. Related Work

Although intensive research has been conducted in the area of cloth and rope simulation for accelerating the simulation [BW98], synthesizing local deformation effects such as buckling effects [CK02], collision detection [THM*03, CTM08, HVS*09], collision handling [BFA02] and learning the physical behaviour to synthesize animation in realtime [dASTH10, WHRO10], few of them have an emphasis on cloth manipulation. Demo animations in cloth simulation are usually limited to movements of clothes worn by human characters, stretching and bending of a piece of cloth, and the movements of a flag or curtains in the wind. In such experiments, the state of the cloth is constant or only passively changing. Our focus is on maneuvers that actively change the state of a cloth with respect to other objects or with itself.

In the rest of this section, we first review computer animation papers that focus on the manipulation of flexible objects. Next, we review papers that deal with folding paper / cloth and knotting ropes in the wider field.

Animation of flexible models

A simple approach for manipulating cloth is to use keyframe animation. Wojtan et al. [WMT06] propose a method to let particles move through the keyframe configurations while keeping the movements plausible. This approach requires a large number of keyframes for creating any animation that involves a lot of close contacts to avoid penetrations, which can be time consuming. Another problem is that there is no abstraction that makes the keyframes applicable for arbitrary pairs of cloth types and objects. Therefore, if the cloth or an object is changed, it will be necessary to edit the keyframes or even insert new keyframes to produce a valid animation.

Igarashi and Hughes [IH02] propose a user interface to put clothes onto characters. The user draws marks on the clothes and on the body. The system then puts the clothes on the body so that the corresponding marks are matched. Although this approach can significantly reduce the effort required for creating static configurations, it does not provide a description about the dynamic manipulation processes such as wrapping or knotting.

Brown et al. [BLM04] propose a method to stably simulate the manipulation process of knotting a rope. The tip particle, called the leader, is controlled to produce knots. The manipulation process is manually created and the maneuver is encoded only by the 3D trajectory of the leader.

Igarashi and Mitani [IM10] provide a 2D interface to specify the layer order and simulate a knotting process. The layered representation is defined in 2D space. It is not convenient for operations such as wrapping, which require direct control in 3D. Ho and Komura [HK09] represent the tangles made between characters by using Gauss Linking numbers. Their method is also limited as they can only represent the relationship of 1D strands.

To summarize, previous researches on manipulation of flexible objects are either limited to direct control and recording of the raw geometry, or have limits on the dimensions they can handle.

Folding cloth / paper and knotting ropes

In the field of Robotics, work has been done on the topic of folding paper [BM08] and cloth [SIT*09, MSCTLA10]. Much of the folding research is built on top of the theories established in the field of computational geometry such as judging whether the crease patterns are foldable [ABD*01] and designing crease patterns to fold a target object [Lan96].

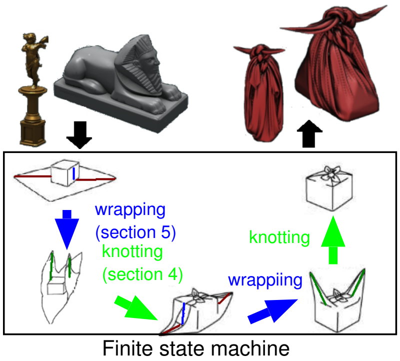


Figure 2: *Workflow of our system. The arrows in the finite state machine represent the transitions betwen the abstract states.*

The folding operations are defined by crease patterns, which are specific to the individual geometry of the paper and cloth. In such representations, there is no room for abstraction that would make them applicable to arbitrary papers or cloths.

Research with abstraction of complex behaviours can be found in robotics research which makes use of knot theory. Several robotics researchers have developed knotting robots that abstract the status of the strands and plan the maneuvers by probabilistic roadmaps [SI07, TMO*06, MTAF06, WAH06]. The roadmap approach is convenient for breaking down the control problem into smaller subproblems. The key points in such an approach are the state representation of the flexible object and the algorithm to convert the abstract state transitions into geometric movements. Previous methods of rope manipulation represent the state based on how the ropes are overlapped with each other when viewing it from a specific direction [DT83], and the state transition is realized by moving the end points. Such a representation requires observing the rope from a specific viewpoint and the control strategy requires imposing multiple constraints [SI07], which can be impractical from the control point of view.

In summary, the roadmap approach can greatly ease the complexity of manipulating cloth. We also apply such an approach in this paper. However, more research is needed for abstraction and bridging the gap between the abstract representation and the geometric instantiation.

Table 1: *The maneuvers for knotting two ropes*

name	***attributes***
Crossing	sign, crossing position, control points
Winding	sign, orientation angle
Tightening	winding/knots

3. Overview

The general design of the system is shown in Fig. 2. Our system produces animation where ropes and cloths are manipulated by following a Finite State Machine (FSM) whose states are connected by discrete maneuvers listed in Table 1 and Table 2.

Among the discrete maneuvers, knotting and wrapping are both guided by Geodesic Control. The Geodesic Control requires defining the control line and the target line, which are both geodesic lines. defined either over a strand, cloth or an object. The basic idea of Geodesic Control is to gradually increase the contact area of the control line and the target line. The target lines are geodesic lines defined over the surface of the target rope for winding, and the surface of the target object for wrapping.

In 2D manifolds, geodesic lines are known as the locally shortest paths between points and are parameterized with constant velocity. They can be computed by extending a straight line on the flattened shape. They may not necessarily be the shortest curve connecting two points, as a curve connecting two points on a sphere by the greater arc is also a geodesic line.

For knotting ropes, the user specifies the part of the rope or cloth that should be used to make the knot, and selects what kind of knots should be made (green transitions in Fig. 2). Knotting is done by conducting a series of winding processes on the specified region of the cloth, which is guided by Geodesic Control. Although arbitrary knots can be synthesized by combining simple winding operations, we have prepared a number of template knots to ease the knotting process. The basics of knotting and how it can be guided by Geodesic Control is explained in Section 4.

Geodesic control can also be applied for manipulating cloth. For wrapping, the user specifies the area of the cloth and the object that is to wrap / be wrapped through our user interface (blue transitions in Fig. 2). Wrapping is guided by gradually increasing the contact area of the control line of the cloth and the target geodesic line defined over the object or its convex hull. We can also knot two ends of cloth as we do with ropes. This process is explained in Section 5.

Using these techniques, examples of controlling ropes and producing wrappings using Geodesic Control are presented in Section 6.

Table 2: *The maneuvers for manipulating cloth*

name	***attributes***
Anchoring	position of cloth and object
Wrapping	control line, geodesic line (target object)
Crossing	control lines, location
Winding	control lines (active/target), direction
Tightening	winding, direction

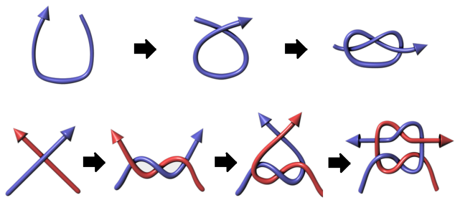
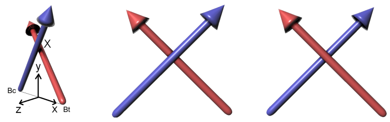


Figure 3: *(top) The process of producing a self-knot by crossing the strand and winding around it. (bottom) The process of producing a granny knot by repeating the crossing and winding.*



(a) (b) (c)

Figure 4: *(a) The state of the two ropes when they are crossed. (b) Positive and (c) negative crosses.*

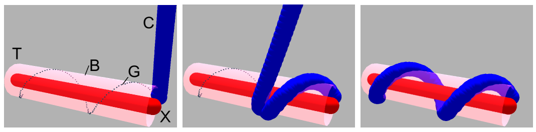


Figure 5: *Snapshots of a blue rope wound around a red rope using Geodesic Control. An bounding tube is produced around the red tube, and a geodesic line is defined on its surface.*

4. Knotting Ropes by Geodesic Control

Most knots in daily life such as self-knots and granny knots can be produced by repeating the process of crossing and winding the ropes. A self-knot can be produced by crossing the strand with itself and conducting a winding (see Fig. 3, top). For a granny knot, after one winding is done, the two ends are raised again, crossed at the middle point, and another winding is conducted (see Fig. 3, bottom).

We automate the knotting process by dividing it into the steps listed below. Let us assume that we wind an active rope denoted by C around a target rope denoted by T. Both ropes are composed of particles connected with rigid rods of constant length.

1. C and T are moved towards each other until a cross is produced at point X.
2. A geodesic line G is computed over T's bounding tube. Starting from X, C is wound around T by gradually increasing the contact area of C and G.
3. The configurations of G and T are updated by optimization, taking into account the physical properties of the ropes, collisions and external forces.
4. Pull the two end of the rope to tighten the winding / knot.

Step 3 is repeated until C and T are wound as desired. Knots are produced by repeating 1-4 for different windings. The individual processes are described below.

4.1. Crossing two ropes

The initial process of winding two ropes is to cross the two ends. Although there can be many ways to produce a cross, our strategy is to lift the two ropes C and T, set a target point X in the middle and bring the ropes towards it. We take this approach because of the following reasons: (1) we want to make a symmetric knot in the middle of the space between the two ends for aesthetic reasons, and (2) more open space will be available to manipulate the rope after lifting because gravity will pull down the rest of the rope.

Let us explain the procedure of crossing the two ropes. A coordinate system is produced as shown in Fig. 4(a)), by using the constrained bottom points of the two ropes and the vertical direction. The two ropes are crossed at a point X, whose x, z coordinates are at the middle of the two bottoms while the height (y coordinate) is set higher than the two bottoms. The two ropes are moved towards X: their z coordinates are slightly adjusted such that they do not collide with each other. A positive cross is produced in the case where a positive wind is to be made, and vice versa for a negative cross (see Fig. 4(b), (c)).

4.2. Winding the two ropes

Here we explain the process of winding the rope C around the rope T. In order to ease the winding process and make the wind / knot appear horizontal and symmetric, T is straightened for a short distance along the x axis before the winding is started.

The winding starts by first computing the bounding tube of T, which is denoted by B, and then gradually attaching C onto the geodesic line G defined on B. (see Fig. 5). The radius of B is set to $r_C + r_T + \varepsilon$, where r_C, r_T are the radius of C and T, respectively, and ε is a small value that is set to $0.1 \times (r_C + r_T)$, such that the two ropes are tightly wound with each other when the particles of T are located on B.

The geodesic line G starts from X_G, the point where the crossing point X is projected onto the bounding tube B. Let us also define the point that is closest to X on curve T as X_T. At X_G, G proceeds in the direction of $\mathbf{n}_X^C$, which is computed by orienting the direction of the rope T at X_T around $\overrightarrow{X_T X_G}$ for θ $(0 < \theta < \frac{\pi}{2})$ (see Fig. 6, left). θ is decided according to how much the user wants to spread the winding across T: a smaller value will make it wider and a larger value will make it tighter (see Fig. 6, right).

G is extended along T until the winding integral [BP06]

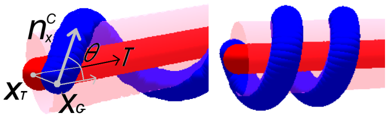


Figure 6: *The orientation of the normal of G with respect to the negative normal of T at X_C (here denoted by θ) determines how much the winding spreads over T (left). A tighter winding produced by a larger θ (right).*

reaches the target value w_0. The winding integral can be computed by

$$\int_{x_0}^{x_1} \frac{\hat{\mathbf{x}} \cdot \mathbf{r}_{GT}(x) \times \mathbf{r}'_{GT}(x)}{|\mathbf{r}_{GT}(x)|^2} dx \quad (1)$$

where x is the direction of winding which is set to the direction of T at X_T, $\hat{\mathbf{x}}$ is the normal vector in the x direction, $[x_0, x_1]$ is the range of computing the winding integral along the x axis, and $\mathbf{r}_{GT}(x)$ is the vector connecting G and T in the $y-z$ plane at each x value (see Fig. 7, left). w_0 is the number of times we want C to wind around T, which is mostly 1 for knots that we make in daily life, such as self-knots and granny knots. The winding integral is equivalent to the Gauss Linking Integral used by Ho and Komura [HK09] but requires only a single integration. It is only applicable when the axis of winding does not turn more than 90 degrees: if that is the case, we need to use the Gauss Linking Integral.

The winding proceeds by iteratively moving each particle p_i^C of C to its target location t_i sampled over G. Starting from X_C, t_i is sampled at every distance r, where r is the distance between every adjacent particle in C. Every time p_i^C reaches its target location t_i, the control switches to the next particle p_{i+1}^C (see Fig. 7, right). We use a constant distance for r, although it is also possible to adaptively subsample more particles in the middle according to the existence of obstacles, and curvature of the other rope, as done in [ST08].

This control strategy for winding is simple and stable. It requires no global path planning as each particle is already in the vicinity of the target location after the previous particle is in contact with G. It also works under a condition that T is dynamically moving. In that case, G is dynamically updated according to the movement of T, and the target locations

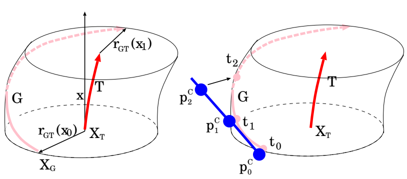


Figure 7: *(left) The winding is computed by integration along the normal vector of T at X_T. (right) C winded around T by gradually attaching particles p_i^C from X_T.*

of the particles are updated. Using the geodesic line as the target line stabilizes the control and the state of the rope as it is the locally shortest path. We will only need to constrain the last particle of p_i^C to preserve the contact of C and T. This is an important feature especially when controlling robots to wind ropes.

4.3. Computing the movements of the control lines

For computing the movements of the ropes, we use an optimization framework that takes into account external force, inextensibility and bending stiffness of the ropes, kinematic constraints and collisions. Technically, we employ an idea similar to the Fast Projection [GHF*07]: we first simulate the movements of the particles composing the rope by only taking into account the gravity, and then, project the updated position of the particles to a manifold that satisfies the constraints by solving an optimization problem. The individual constraints and the optimization process is explained below.

Inextensibility constraints: A control line or a rope is modelled by particles whose positions are $\mathbf{p}_i$ where $i \in [0 : n-1]$ and links between them whose length are r. The inextensibility constraints can be formed as:

$$C_e = \| \mathbf{p}_{i+1} - \mathbf{p}_i \|^2 - r^2 = 0. \quad (2)$$

Bending stiffness: For bending stiffness, we reformulate the idea used in [Jak01] as below:

$$C_b = \| \mathbf{p}_{i+1} - \mathbf{p}_{i-1} \| - 2r = 0. \quad (3)$$

Kinematic constraints: The active rope is controlled by moving one of its particles p^c towards its target location $\mathbf{p}_t$. As $\mathbf{p}_t$ can be too far away from p_c to be reached by one step, a series of intermediate target points $\mathbf{p}_k = \frac{k}{n_c - k}\mathbf{p}_c + \frac{1}{n_c}\mathbf{p}_t (k \in [1 : n_c - 1]$ are defined, where n_c is a constant integer defined by the initial distance between $\mathbf{p}_c$ and $\mathbf{p}_t$. At every iteration, the following kinematic constraint is imposed:

$$C_k = \| \mathbf{p}_c - \mathbf{p}_k \| = 0. \quad (4)$$

Collision constraints: We model ropes by linked capsules. The collisions between two capsules s_i and s_j can be resolved by imposing the following constraint:

$$C_c^{s_i,s_j} = D(Capsule_{s_i}, Capsule_{s_j}) - (r_{s_i} + r_{s_j}) >= 0 \quad (5)$$

where $D()$ returns the shortest distance between the two capsules, and r_{s_i}, r_{s_j} are the radii of the two capsules, whose values are same as r_C and r_T. We do not apply this constraint to adjacent capsules in each rope as their ends are already overlapping to compose the rope.

Projection to the manifold of constraints: After adjusting the particle locations of the control ropes based on the gravity, their positions are mapped on to the manifold of the constraints defined above. This is done by computing the minimum particle displacement $\Delta\mathbf{p} = (\Delta\mathbf{p}_1^\top, ..., \Delta\mathbf{p}_n^\top)^\top$

needed for satisfying all the constraints. As the inextensibility constraints, bending stiffness and collision constraints are nonlinear, we linearize them by the finite difference method. The collision constraints are handled as hard constraints as they are essential for preventing penetrations, which are visually influential. The rest are handled as soft constraints to increase the adaptiveness of the ropes. $\Delta\mathbf{p}$ is computed by solving the following optmization problem:

$$\min_{\Delta\mathbf{p}} \frac{1}{2}\Delta\mathbf{p}^\intercal\Delta\mathbf{p} + \frac{1}{2}a||\mathbf{J}_e\Delta\mathbf{p} - \alpha||^2 + \frac{1}{2}b||\mathbf{J}_b\Delta\mathbf{p} - \beta||^2 + \frac{1}{2}c||\mathbf{J}_k\Delta\mathbf{p} - \gamma||^2 \quad (6)$$

subject to

$$\mathbf{J}_c\Delta\mathbf{p} + \sigma = \mathbf{0} \qquad \text{only imposed if} \quad \sigma < 0 \quad (7)$$

where $(\alpha, \beta, \gamma, \sigma)$ and $(\mathbf{J}_e, \mathbf{J}_b, \mathbf{J}_k, \mathbf{J}_c)$ are the values and Jacobians of (C_e, C_b, C_k, C_c), respectively. and a, b and c are coefficients where $a = exp(abs(\alpha) \times 10), b = 2000, c = 3000$.

A solution can be obtained by solving the following sparse linear equation:

$$\begin{pmatrix} \mathbf{I} + \mathbf{J}_e^\intercal\mathbf{J}_e + \mathbf{J}_b^\intercal\mathbf{J}_b + \mathbf{J}_k^\intercal\mathbf{J}_k & \mathbf{J}_c^\intercal \\ \mathbf{J}_c & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{p} \\ \lambda \end{pmatrix} = \begin{pmatrix} a\mathbf{J}_e^\intercal\alpha + b\mathbf{J}_b^\intercal\beta + c\mathbf{J}_k^\intercal\gamma \\ -\sigma \end{pmatrix} \quad (8)$$

After $\Delta\mathbf{p}$ is computed, the particle locations are updated by adding the corresponding elements to $\mathbf{p}_i$. This procedure can be iterated until all the constraints are satified, although we find the first iteration usually produces satisfactory results.

5. Controlling Cloth by Geodesic Control

The idea of Geodesic Control explained in the previous section can be applied for controlling cloth to wrap it around an object or knotting two ends of cloth. We explain about these processes in this section. Knotting is done in the same way as ropes, but with an additional procedure based on medial axes to reduce the self-penetrations between the cloth.

5.1. Wrapping by Geodesic Control

Each wrapping maneuver is achieved by gradually increasing the contact area of the control line of the cloth with the target curve defined over the object or its convex hull. The procedure to compute these lines and control the cloth using them is first explained. The movement of the rest of the cloth is computed passively using a generic simulator based on particle physics. This process is explained next.

Guiding wrapping by control lines: We now define a control line and a target line on the cloth and object, respectively to guide the cloth to wrap around the object by Geodesic Control. The control lines are straight lines over the cloth defined by the user. In most cases, they are lines connecting the center and the corners of the cloth. The target lines are geodesic lines defined either over the wrapped object or its convex hull, depending on the wrapping style. In the initial configuration, the control line must be in contact with the target line (see Fig. 8, left). Either the object must be shifted over the cloth or the target curve needs to be redefined if this condition is not met.

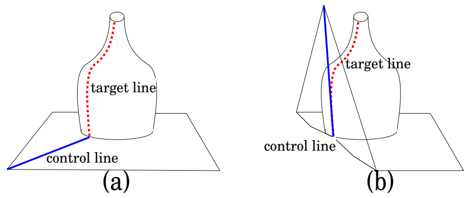


Figure 8: *(a) The control line on the cloth and the target line on the object. (b) The cloth wrapped around the target object by overlapping the control line on the target line.*

As the degrees of freedom of a cloth is much greater than that of a rope, it is necessary to further constrain it to make it robustly wrap around the object. This is achieved by straining the cloth in the direction of the control line, which eases the process of overlapping the control and the target line.

After the cloth is strained, the control line is overlapped with the target line by bending it at the edge of the object that the control line is in contact with (see Fig. 8, right). The control line is rotated until it fully contacts the target curve on the polygon that includes the last contact point. This process is repeated until the control line fully overlaps with the target line. In practice, the cloth is manipulated by the series of particles defined over the control line. Their movements are updated by solving the optimization problem described in Section 4.3.

Cloth movements by particle physics: Once the movement of the control lines are computed, we constrain the position of the particles that compose the control lines and compute the movements of the rest of the cloth particles by a generic simulator based on particle physics. Regarding the physical properties, we take into account the stretchiness and the bending stiffness between the particles, whose values are set to 0.9 and 0.011. The distance between the particles in the grid are set to 0.2 and the thickness of the cloth is set to 0.3, which produces a bounding sphere whose radius is 0.15 for collision detection. We use PhysX 2.84 [NVi] for the physical simulator.

5.2. Knotting Cloth

For knotting cloth, we apply the Geodesic Control to the control lines that are defined on the surface of the cloth. Similar to the process of winding ropes, we define the active line and the target line. A bounding tube of a constant radius is

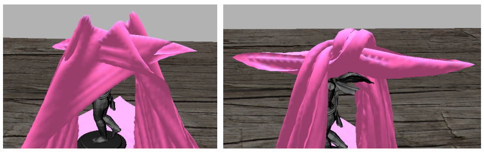
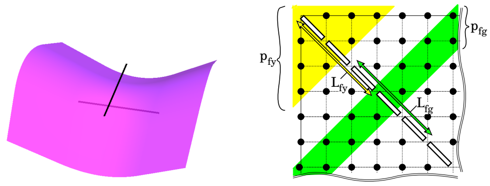


Figure 9: *Self-penetrations happening during knotting two ends of a cloth by a generic simulator (left). They can be greatly eliminated by separating the ends by the medial axis (right).*



(a) (b)

Figure 10: *(a) A visualization of the medial sheet computed from the configuration of the two control lines. It is to be noted that we do not explicitly compute the medial sheet. (b) The enlarged top-left corner of the cloth : The free particles in the yellow (p_{fy}) and green (p_{fg}) region are associated to the line segments specified by the yellow (L_{fy}) and green arrows (L_{fg}).*

defined around the target curve, and a geodesic line is computed on the surface of the tube. Finally, the active curve is wound around the target curve by gradually locating the particles of the active curve around the bounding tube. The tube is defined only for a short distance around the area where the winding is going to be made. This is to avoid a large area of the cloth to be squeezed. The radius of the tube is computed based on the amount of cloth involved in the knotting. We also add an extra post-processing stage to minimize the amount of cloth-cloth penetration. These processes are explained below.

Reducing self-penetration using the medial axis: Self-collision is a serious issue for particle-based cloth simulators. The artifacts are evident when simulating complex interactions such as knotting using generic simulators that use naive collison detection approaches (see Fig. 9, left). Although its effect can be reduced using asynchronous stepping as done in [HVS*09], this requires a huge amount of computation, which may not be acceptable for real-time applications. Here we eliminate such artifacts by inserting an additional medial axis layer (medial sheet) between the two control lines (see Fig. 10,(a)) and forcing the associated particles to stay on the same side of the medial sheet.

Each free particle f is associated to a region of the control line composed of M line segments, which is denoted here by $\mathbf{L_f}$. $\mathbf{L_f}$ is the closest set of line segments from f along the surface of the cloth that is composed of M line segments (M is set to 3 in our examples). Fig. 10,(b) shows an example of free particles and their associated line segments.

After computing $\mathbf{p}_f$, which is the position of particle f, using the physical simulator while constraining the control lines, we examine the distance between f and all the line segments; if the closest particle is not included in $\mathbf{L_f}$, it is moved towards the closest point on $\mathbf{L_f}$. More specifically, let us denote by d_f^i the distance between f and all the line segments $l_i (i \in [1:m])$. If $\arg\min_i d_f^i (1 \le i \le m) \notin \mathbf{L_f}$, we move $\mathbf{p}_f$ towards its projection point on $\mathbf{L_f}$, which is defined here by $\mathbf{p}_f^p$. Namely, $\mathbf{p}_f$ is updated by $\mathbf{p}_f \leftarrow \mathbf{p}_f + a \frac{\mathbf{p}_f^p - \mathbf{p}_f}{||\mathbf{p}_f^p - \mathbf{p}_f||}$ where a = 0.01. This process is applied for all the free particles. Then, we run the simulation again while constraining all the $\mathbf{p}_f$'s that have been updated. This process is repeated until either each free particle is closest to its associated line segment, i.e., located within the Voronoi regions of its associated line segment, or the maximum number of iterations is reached, which is set to 3 in our experiments.

Our method can significantly reduce the penetrations between the two cloth-piece during the knotting procedure (see Fig. 9, right). The twisted medial surface also helps to produce a natural effect of a cloth being squeezed.

6. Experimental Results

In this section, we first show an example of rope winding. Next, we show examples where various furoshiki wraps are generated by sequentially applying the winding and knotting techniques. The readers are referred to the supplementary video for further details of the folding process. Finally, we explain about the computational costs.

6.1. Rope winding

We first show an example of winding one rope around another moving rope in an environment with multiple obstacles. Snapshots of the scene are shown in Fig. 11. This is an example that is difficult even using manual control as it is necessary to avoid the obstacles and unnecessary winding while following and winding around the moving rope. Geodesic control can successfully achieve the task without global path planning.

6.2. Furoshiki wrapping

We wrap objects shown in Fig. 12(a) by a two granny-knot box wrapping, a wine-bottle wrapping and a basket wrapping.

Two granny-knot box wrapping: A two granny-knot box wrapping is produced by repeating the process of wrapping and making a granny-knot at the top of the object using the

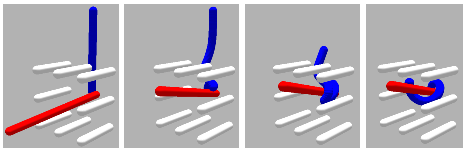


Figure 11: *Winding one rope around another dynamically moving rope using Geodesic Control in an environment with obstacles.*

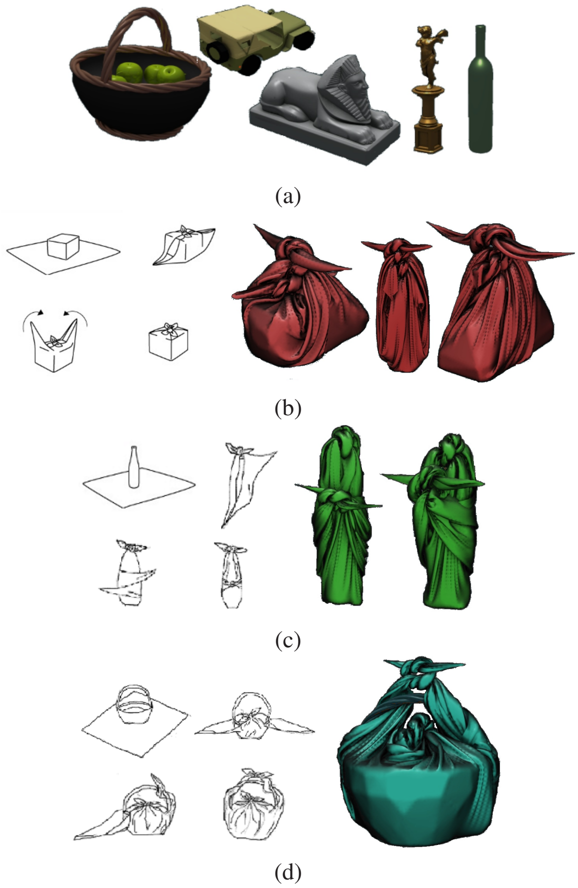


Figure 12: *(a) The objects that are wrapped in the furoshiki wrapping experiment. (b) The FSM of box wrapping (left) and the jeep, cupid and sphinx wrapped by this style (right). (c) The FSM of wine bottle wrapping (left) and a bottle and cupid wrapped by this style(right). (d) The FSM of basket wrapping (left) and a basket wrapped by this style(right).*

opposite corners of the cloth. The procedure of the wrapping can be found in Fig. 12(b), left and Table 4. We show results of wrapping a jeep model, a sphinx sculpture and a cupid sculpture, which all have different structures. Despite such variations, the objects are successfully wrapped as shown in Fig. 12(b), right.

Wine bottle wrapping: The wine bottle wrapping is suitable for wrapping cylindrical objects, such as a wine bottle. The procedure of the wrapping can be found in Fig. 12(c), left and Table 5. This wrapping involves winding the cloth around the bottle using Geodesic Control. We show examples of wrapping a wine bottle and the cupid structure in Fig. 12(c), right.

Basket Wrapping Finally, a basket wrapping was done to a basket. The procedure of the wrapping can be found in Fig. 12(d), left and Table 6. In this example, the cloth winds around the handle, which is also guided by Geodesic Control. The result is shown in Fig. 12(d), right.

6.3. Computational Costs

Regarding the complexity of our method, the set-up and solving of Eq.6 is $O(m)$, computation of the medial axis and updating the particle positions is $O(m \times n)$, and the particle simulation of the cloth based on a generic simulator is $O(n)$, where m is the number of rods on the control line and n is the number of particles composing the cloth.

The breakdown of the computation of each step in the descending order is collision resolving by the medial axis, (0.2 second per frame), the set-up and solving of Eq.6 (0.016spf) and particle simulation (0.001spf). Note that we only apply a naive approach that computes the distance of every pair of particles and line segments for the collision resolving. An acceleration can be achieved by applying methods such as oct-trees.

The cloth used in the system is composed of 26569 (163×163) particles, and we can generate an animation of 1 second physical simulation in roughly 3 seconds, using a system with a Intel Core i7 2.93GHz CPU, 6GB RAM and NVidia GeForce GTS 240. The computation for one frame is completed in around 0.22 seconds. We used UMFPACK for solving the optimization problem and PhysX for the cloth simulation.

7. Discussions

One of the advantages of our method is that the complex maneuvers of wrapping and knotting are highly abstracted. Therefore, the description of the maneuvers are applicable to different combinations of objects and pieces of cloth, as far as the size of the cloth affords it.

Successful winding can be easily achieved by Geodesic Control, even under conditions where there are multiple obstacles or the target is dynamically moving, which can be difficult for other methods such as Topology Coordinates [HK09] or "follow the leader" [BLM04]. The Topology Coordinates is another local control method that can be applied for winding motion. If there are obstacles that the rope should not wind around, the user needs to specify that as additional constraints. When there are multiple obstacles, such extra constraints can cause local minima. The follow the leader approach requires global path planning of the rope tips and there is a chance that the loose area gets hooked with obstacles in the environment. Geodesic Control, however, can robustly wind around another rope or object without any costly computation. The initial crossing is the only condition for starting the control, but that is not difficult, especially for tasks in which the ropes are open ended.

Geodesic Control is a local control solution for winding and wrapping procedures, and it is not a solution that can replace global path planning. In our examples, the FSM takes on the role of global path planning. For maneuvers such as passing the rope between a series of ropes or a narrow area, a global path planner that provides the target geodesic lines to follow is needed. One possible extension is to set up a FSM that is based on homotopy groups [BLK11]. The homotopy groups categorize paths based on the topology of the paths. Once a homotopy group is given, we can compute the geodesic path within that homotopy group.

There are several ad-hoc steps in our system. Some parameters of the each manuever are manually specified or set by simple ad-hoc rules. For example, the geodesic lines on the objects (the target lines) are manually defined by the user. The user specifies the position of the end point and then a geodesic line is computed by finding the shortest route connecting the two end points. One problem of this approach is that there are cases where the geodesic line takes a different route from that which is desired (such as taking the shorter arc on a sphere instead of the desired longer arc). One way to solve this problem is to let the user also specify the tangent vector at one of the end points, compute the geodesic curve based on this, and finally let the system adjust it such that it passes through the other end point. Automatically deciding parameter values such as the position of the knot based on the geometry of the target object, the length of the control line and the style of the wrap is a challenging but interesting topic to work on.

Our approach can also be applied to overlap control lines to non-geodesic lines defined on objects. As mentioned earlier, the advantage of overlapping the control curve with geodesic lines of the object or its convex hull is in its physical stability; only the last control point needs to be constrained to keep the same configuration. For animation purposes, this condition can be dropped; an interface to letting users draw curves on surfaces and allowing flexible objects follow them for modeling objects and creating animation would be an interesting extension.

8. Conclusion and Future Work

In this paper, we have proposed a method to synthesize animations of wrapping and knotting cloths by introducing discrete representations and control methods to interpolate such representations. Currently, the movement of the flexible object is guided by the control curves. For animation and control purposes, it is necessary to realize such movements through the interaction of the robot / character with the flexible object. This is an interesting future direction for research.

Acknowledgement

The authors thank the anonymous reviewers for their constructive comments, Hubert P.H. Shum, Edmond S.L. Ho, Adam Barnett and Chiew-Lan Tai for their valuable feedback and Rami Ali Al-ashqar for helping to produce the video. This work was partly funded by EU IST-FP7-IP TOMSY (Ref:270436), EPSRC (EP/H012338/1) and the China Scholarship Council.

References

[ABD*01] ARKIN E. M., BENDER M. A., DEMAINE E. D., DEMAINE M. L., MITCHELL J. S. B., SETHIA S., SKIENA S. S.: When can you fold a map? *Proc of WADS 2001 2125* (2001), 401–413.

[BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans Graph 21*, 3 (2002), 594–603.

[BLK11] BHATTACHARYA S., LIKHACHEV M., KUMAR V.: Identification and representation of homotopy classes of trajectories for search-based path planning in 3d. *Proc of Robotics: Science and Systems* (2011).

[BLM04] BROWN J., LATOMBE J.-C., MONTGOMERY K.: Real-time knot tying simulation,. *The Visual Computer 20*, 2-3 (2004), 165–179.

[BM08] BALKCOM D., MASON M.: Robotic origami folding. *International Journal of Robotics Research* (2008), 613–627.

[BP06] BERGER M. A., PRIOR C.: The writhe of open and closed curves. *Journal of Physics A: Mathematical and General 39*, 26 (2006).

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. *Computer Graphics (Proc of SIGGRAPH)* (1998), 43–54.

[CK02] CHOI K.-J., KO H.-S.: Stable but responsive cloth. *ACM Trans Graph 21*, 3 (2002), 604–611.

[CTM08] CURTIS S., TAMSTORF R., MANOCHA D.: Fast collision detection for deformable models using representative-triangles. In *Proc. of Symp on Interactive 3D graphics and games* (2008), pp. 61–69.

[dASTH10] DE AGUIAR E., SIGAL L., TREUILLE A., HODGINS J. K.: Stable spaces for real-time clothing. *ACM Trans Graph 29* (2010), 106:1–106:9.

[DT83] DOWKER C. H., THISTLETHWAITE M. B.: Classification of knot projections. *Topology and its Applications 16*, 1 (1983), 19–31.

[GHF*07] GOLDENTHAL R., HARMON D., FATTAL R., BERCOVIER M., GRINSPUN E.: Efficient simulation of inextensible cloth. *ACM Trans Graph 26*, 3 (2007), 49.

[HK09] HO E. S. L., KOMURA T.: Character motion synthesis by topology coordinates. *Computer Graphics Forum (Proc of Eurographics 2009) 28*, 2 (2009).

[HVS*09] HARMON D., VOUGA E., SMITH B., TAMSTORF R., GRINSPUN E.: Asynchronous Contact Mechanics. *ACM Trans Graph 28*, 3 (Aug 2009).

[IH02] IGARASHI T., HUGHES J. F.: Clothing manipulation. In *UIST '02: Proc of ACM Symp on User Interface Software and Technology* (2002), pp. 91–100.

[IM10] IGARASHI T., MITANI J.: Apparent layer operations for the manipulation of deformable objects. *ACM Trans Graph 29*, 4 (2010).

[Jak01] JAKOBSEN T.: Advanced character physics. *In Game Developers Conference Proc* (2001), 383–401.

[Lan96] LANG R. J.: A computational algorithm for origami design. In *Proc of Symp on Computational Geometry* (1996), pp. 98–105.

[MSCTLA10] MAITIN-SHEPARD J., CUSUMANO-TOWNER M., LEI J., ABBEEL P.: Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. *Proc of IEEE Int Conf on Robotics and Automation* (2010).

[MTAF06] MATSUNO T., TAMAKI D., ARAI F., FUKUDA T.: Manipulation of deformable linear objects using knot invariants to classify the object condition based on image sensor information. *IEEE/ASME Trans Mechatronics 11*, 4 (2006).

[NVi] NVIDIA: Physx. *http://www.nvidia.com/object/physx_new.html*.

[SI07] SAHA M., ISTO P.: Manipulation planning for deformable linear objects. *IEEE Trans Robotics 23* (2007), 1141–1150.

[SIT*09] SUGIURA Y., IGARASHI T., TAKAHASHI H., GOWON T. A., FERNANDO C. L., SUGIMOTO M., INAMI M.: Graphical instruction for a garment folding robot. *ACM SIGGRAPH Emerging Technologies* (2009).

[ST08] SPILLMANN J., TESCHNER M.: An adaptive contact model for the robust simulation of knots. *Computer Graphics Forum (Proc of Eurographics 2008) 27*, 2 (2008), 497–506.

[THM*03] TESCHNER M., HEIDELBERGER B., MUELLER M., POMERANETS D., GROSS M.: Optimized spatial hashing for collision detection of deformable objects. pp. 47–54.

[TMO*06] TAKAMATSU J., MORITA T., OGAWARA K., KIMURA H., IKEUCHI K.: Representation for knot-tying tasks,. *IEEE Trans Robotics 22*, 1 (2006), 65–78.

[WAH06] WAKAMATSU H., ARAI E., HIRAI S.: Knotting/unknotting manipulation of deformable linear objects. *Int. J. Rob. Res. 25*, 4 (2006), 371–395.

[WHRO10] WANG H., HECHT F., RAMAMOORTHI R., O'BRIEN J.: Example-based wrinkle synthesis for clothing animation. *ACM Trans Graph 29*, 4 (2010), 107:1–107:8.

[WMT06] WOJTAN C., MUCHA P. J., TURK G.: Keyframe control of complex particle systems using the adjoint method. In *Proc of ACM SIGGRAPH/Eurographics Symp on Computer Animation* (2006), pp. 15–23.

Appendix

Here we show the sequence of maneuvers of the FSMs for the two granny-knot box wrapping (Table 4), wine bottle wrapping (Table 5) and and basket wrapping (Table 6). The identities of corners, area and control lines are defined as shown in Fig. 13. For the sake of simplicity, we first define the granny-knot maneuver which we used for all our demos. The maneuver takes two control lines and the knotting position as input, and produce a granny knot.

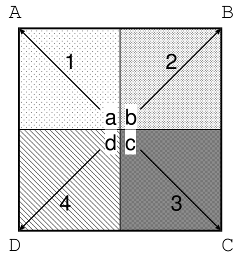


Figure 13: *The identities of the corners, area and control lines defined for our experimental results.*

Table 3: *The maneuvers and attributes of GrannyKnot*

	maneuver	***attributes***
1	Crossing	control line C and T, top of object center
2	Winding	control line C around T, top of object center, negative
3	Tightening	winding made at step 2, towards center
4	Crossing	control line T and C, top of winding made at step 3
5	Winding	control line T around C, top of winding made at step 3, negative
6	Tightening	winding made at step 4, towards winding made at step 5

Table 4: *The maneuvers and attributes of two granny-knot box wrapping*

	maneuver	***attributes***
1	Anchoring	center
2	Wrapping	corner A and C, area a and c
3	GrannyKnot	control line 1 and 3,top of object center
4	Wrapping	corner B and D, area b and d.
5	GrannyKnot	control line 2 and 4, top of knot made at step 3

Table 5: *The maneuvers and attributes of wine bottle wrapping box wrapping*

	maneuver	***attributes***
1	Anchoring	center
2	Wrapping	corner A,C and area a,c
3	GrannyKnot	control line 1 and 3, top of object center
4	Wrapping	corner B and area b
5	Wrapping	corner D, and area d
6	GrannyKnot	control line 2 and 4, front of object

Table 6: *The maneuvers and attributes of a basket wrapping.*

	maneuver	***attributes***
1	Anchoring	corner A
2	Wrapping	corner A,C and area a,c
3	GrannyKnot	control line 1,3, top of bottom half of basket
4	Winding	control line 2, basket handle, negative
5	Winding	control line 4, basket handle, negative
6	GrannyKnot	control line 2,4, top of basket handle