Dynamic Axial Curve-pair based Deformation and its Application

香港中文大學圖書館系統 THE CHINESE UNIVERSITY OF HONG KONG 0 5 JAN 2010 UNIVERSITY LIBRARY SYSTEM

CHAN, Man Leung Dunco

A Thesis Submitted in Partial Fulfillment
of the Requirement for the Degree of
Master of Philosophy
in
Automation and Computer-Aided Engineering

© The Chinese University of Hong Kong
Nov 2008

The Chinese University of Hong Kong holds the copyright of this thesis. Any person(s) intending to use a part or a whole of the materials in the thesis in a proposed publication must seek copyright release from the Dean of the Graduate School

香港中文大學
圖書館系統
★ ★
THE CHINESE UNIVERSITY OF HONG KONG
0 5 JAN 2010
UNIVERSITY
LIBRARY SYSTEM

Thesis/Assessment Committee

Professor Chung, Chi-kit Ronald (Chair)
Professor Hui, Kin-chuen (Thesis Supervisor)
Professor Wang, Changling Charlie (Committee Member)
Professor Lau, Wing-hung Rynson (External Examiner)

Abstract

Deformation of 3D objects plays an important role in computer graphics, computer simulation and computer-aided design. Using a deformation tool, a model is allowed to take useful and intuitive shapes. The axial deformation technique allows a 3D object to be deformed by adjusting the shape of an axial curve. However, due to lack of control on the local coordinate frame, unexpected twist may result. The axial curve-pair based deformation technique proposed a scheme which allows the local coordinate frame to be controlled intuitively. Nevertheless, performing a deformation of an object using these techniques could be tedious because designers are required to interactively adjust a number of parameters, control points, to achieve a desired shape. The dynamic axial curve-pair based deformation technique enhances the technique by incorporating physics based deformation in the deformation process. A special designed mass-spring model for the special curve-pair structure is adopted to model an elastic axial curve-pair skeleton. Physics-based deformation performed on the point masses of the mass spring system deforms the embedded curve-pair. An object model could come to live if a kinetic artistic pose is applied to the object. This technique is particularly useful for physics-based deformation of plants, animals and characters. More intuitive designs could be guaranteed when a deformation is performed taking into consideration the effect of forces and geometric constraints. In addition, the proposed technique reduces the number of parameters to be manipulated in a deformation and hence provides a more intuitive interface for the design of freeform objects.

摘要

在今時今日,三維空間物件之變形在計算機圖像處理、計算機模擬及計算機輔助設計幾個範疇中佔有相當重要的地位。在一些變形工具之輔助下, 設計師可以任意去塑造他們心目中所需要之模型外形。而在其中,軸向變形技術(Axial Curve Deformation)容許設計師通過把軸心變形而達到改變嵌入物形態之目的。然而在變形過程中,設計師沒辦法完全控制其地方座標徑(Local Coordinate Frame)之訂定,引至有嵌入物產生不良變形之現象。而雙曲線軸心變形技術(Axial Curve-pair based Deformation)則彌補了這方面的缺陷。

誠然,這些工具都需要設計師去控制大量之變形控制點。因此我們在雙曲線軸心變形技術之基層上,提出了通過加入力學參數去實現符合物理表現之變形結果。一個特別的質量彈簧系統(Mass-spring System)將會使用去模擬雙曲線軸心之變形,亦因為物理參數之介入,嵌入物將擁有更多具動感之形態。此技術能減省設計師在塑造新物品外形的時間,並把設計過程提昇到可運用物理參數以造出更多有趣之物件外形。

Acknowledgement

I sincerely thank my supervisor, Professor Kin-chuen Hui of the Department of Automation and Computer-Aided Engineering, the Chinese University of Hong Kong for giving me an opportunity to work at the CAD Lab and his continuous support in the Master of Philosophy program.

I also thank the Department of Automation and Computer-Aided Engineering of the Chinese University of Hong Kong for providing me a brilliant working environment and professional facilities throughout my research. And the last but not the least, I would like to thank my colleagues in the CAD Lab for their support and snacks during my research time.

Content

List of figures

Chapter 1

Introduction

1.1 Background

Deformation of 3D objects plays an important role in computer graphics, computer simulation and computer-aided design. Generally, deformation can be regarded as a mapping from R^3 to itself except Bechmann [29] works in R^n. All the deformation methods independent of the representation of the underlying objects can be divided into two classes according to whether it requires a deformation tool or not [30].

Barr's global and local deformation [31] and the space deformation introduced by Borrel and Bechmann [30] are two kinds of deformation methods which does not require the use of deformation tools. In this method, deformations are usually represented as matrix of functions or constraints of the location of selected space points. A large range of deformed shape such as arbitrary shaped bump can be modeled using these techniques.

The other class of deformation methods requires the use of some deformation tools. Representative methods include Axial Deformation [8] and Free-form deformation (FFD) [1]. Both methods establish a mapping between the global object space and a local parametric space determined by a deformation tool. Using a deformation tool, a geometric model can be deformed to take meaningful and creative shapes intuitively.

Model concepts come to live if a living poses and motions are applied to the model. In kinetic art, artistic poses and motions are a result of deformation of object shape.

The deformation method which requires some deformation tools embeds an object in a space which can be warped and hence deforms the embedded object. By modifying the shape of existing models, a new design can be created. From the user's point of view, this method is very effective.

Taking the advantage of different deformation tools, we are looking for a new approach which provides a more intuitive mean for controlling the deformation.

1.2 Prior work

The free-form deformations (FFD) technique introduced by Sederberg and Parry [1] is a popular tool for modeling and animating deformable objects. The original lattice of the FFD consists of control points along three mutually perpendicular axes, which defines a space of Brezier volume. However, creating animation using the FFD could be tedious because a large number of control points are required to be manipulated.

Borrel [2] proposed a space deformation model which does not require the use of lattices. A different approach is to control FFD using parametric curve and surfaces [7]. This technique simplifies the task by reducing the numbers of control points to be manipulated. There are other works aiming at introducing a dynamic setting in a deformation tool in order to minimize the work load in handling the control lattice.

Another kind of deformation tool, the axial deformation technique, was introduced by Lazarus et al. [8] which allows the shape of a 3D objects to be deformed by adjusting the shape of an axial curve. New design idea is obtained by deforming existing geometric shapes. However, due to lack of control on the local coordinate frame, unexpected twist of object may be obtained.

There has been substantial interest of late in developing intuitive interactive techniques such as deformation by painting over surfaces [33], WIRES [26] and sketch mesh deformation [32].

In [3], dynamic deformable surfaces as well as a dynamic deformation technique using FFD are modeled using NURBS. In [4], the free-form deformations technique is used to model muscle deformation. FFD lattices are attached to a skeleton. Dynamic motion governed by a mass-spring system deforms the lattices. In [5], a technique is proposed to synthesize the motions of animated characters modeled as 3D mass-spring lattices. In [6], the dynamic FFD technique which makes use of deformation modes is presented for the animation of deformable characters. These works indicate a dynamic configuration could eventually extend a deformation tools to become a more effective one.

However, the dynamic FFD techniques in these works could not guarantee a smooth bending or twisting deformation. And the dynamics configuration is applied to the control lattice instead of the embedded object. Furthermore, the number of degrees of freedoms to be manipulated is still large compared with a more compact representation, the axial deformation technique, in defining an implicit global deformation.

Hui [9] proposed a free-form design method using axial curve-pairs. A curve-pair composes of a primary and an orientation curve allows the local coordinate frame to be controlled intuitively. Thus, by associating 3D objects to the curve-pair, these objects can be stretched, bended and twisted through manipulating the curve-pair.

The axial curve-pair deformation technique is effective for manipulating complex shapes in industrial and aesthetic design and flexible models such as characters and animals. However, users experience the same problems mentioned in previous paragraphs. Control points of curve-pair, are still required to be adjusted in order to create the desired shapes.

1.3 Objectives

The axial curve-pair based deformation technique is a good basis of further researches. Knowing that dynamic deformation tool can be used to create physically meaningful kinetic and artistic poses of 3D shapes as well as minimizing user interactions, this research proposes a configuration to extend the axial curve-pair deformation technique to a dynamic environment. Since a physics-based object design approach allows models to be deformed by dynamically applying forces instead of relying on adjusting control points, this technique provides a more intuitive approach for creating physically meaningful design.

There are a number of problems to be tackled ahead:

- What formulation of dynamics is suitable for the required deformation?
- How to implement such dynamic formulation?
- How could such model provide a more intuitive mean for controlling deformation?

Qin and Terzopoulos [10] introduced a physics-based framework for geometric design with NURBS known as Dynamic NURBS. In their work, D-NURBS curves, tensor-product D-NURBS surface, swung D-NURBS surfaces, and the triangular D-NURBS surfaces are formulated.

However, the curve-pair representation of the axial curve-pair deformation technique is not a simple relationship between two NURBS curves. The two curves have different functionalities. The bending and stretching properties are expressed by adjusting the primary curve while the twisting property is expressed by rotating the orientation curve about the primary curve. When the primary curve is modified, the orientation curve has to be modified correspondingly and automatically. The offset direction between the primary curve and the orientation curve has to be kept consistent when the shape and twist of the curve-pair are adjusted.

The curve-pair is then required to be manipulated as a pair. However, there is no exact mathematical representation for a curve-pair model. Because of its unique geometric representation, the formulation of D-NURBS curves or D-NURBS surface cannot be applied to the curve-pair model. These will be further discussed in Section 3.1.

In [11][12][13][14][15], deformable objects are animated using mass-spring models. In [4], the free-form deformations technique is used to model the muscle deformation. FFD lattices are attached to the skeleton where dynamic motion governed by a mass-spring system deforms the lattices. In [5], the work targets to synthesize the motion of animated characters modeled as 3D mass-spring lattices. These works applied a dynamic deformation technique directly to the lattice and produce a faster simulation and more efficiently comparing other approaches in which dynamics is applied directly to the geometry of object. However, applying dynamics directly to the lattice does not guarantee a desire deformation on the embedded object. The same problem remains if we apply the dynamics to the control points of curve-pair. Moreover, a spring mass system with inter-lattice linear and torsional springs applied to the curve-pair does not have a restoring ability. These will be discussed in Chapter 3.

As a result, we want to extend the axial curve-pair deformation technique to achieve a physics-based deformation effect by fitting the curve-pair to a special mass spring system with shape restoring ability. This technique aims to provide an effective tool for shape design. An overview of this technique will be discussed in the next section.

1.4 Proposed method

Our representation of dynamic axial curve-pair deformation makes use of an energy model, the mass-spring system. A curve-pair is fitted to the system to emulate the physical properties.

A number of physical properties are incorporated into the system. They are stretching, bending and twisting. These properties are emulated by linear and torsional springs (Figure 1.1). Each point mass is linked to its neighbors by springs representing the different physical behaviors of the curve-pair.

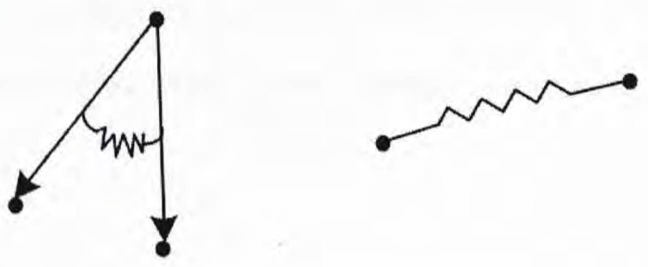


Figure 1.1 - The metric spring (right) and the torsional spring (left)

The system is then solved by using a numerical integrator. The method adopted is the forth order Runge-Kutta method with adaptive step control. An adaptive step control minimizes the computational complexity as well as stabilizes the system with large speed variations.

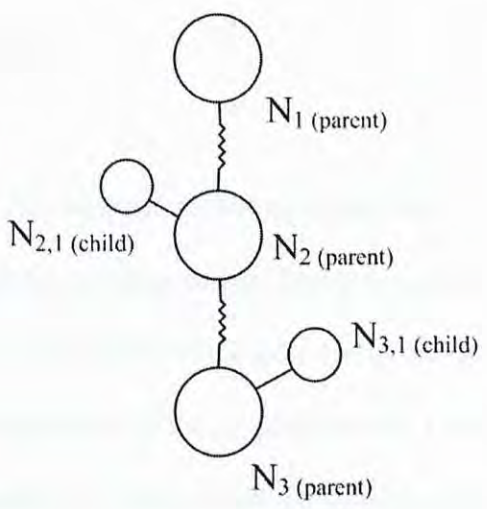


Figure 1.2 - An overview of a dynamic hierarchical representation

Using a hierarchy of axial curve-pairs (Figure 1.2), different deformation effects can be achieved. A common approach to represent hierarchical relation is to use a parent and child relationship. In our approach, besides the geometric relation of curve-pairs, torsional springs are also used to connect curve-pairs. As a result, the connected axial curve-pairs may affect each others. In other words, a local deformation may result in a global deformation.

The physical deformation of objects also involves another important issue: collision detection. In our work, we adopted the axial bounding cylinder method. A bounding cylinder enclosing the deforming object is used for collision tests.

1.5 Thesis outline

This thesis is composed of five chapters including 1) Introduction, 2) Axial curve-pair based deformation, 3) Dynamic axial curve-pair based deformation, 4) Implementation and experimental results and 5) Conclusion. In Chapter 1, the background of the research will be addressed as well as a description of the proposed work. In Chapter 2, the basis of the proposed work, the axial curve-pair deformation technique, will be reviewed. The details of the dynamic formulation in addition to the axial curve-pair deformation technique will be presented in Chapter 3. In Chapter 4, the implementation and the experiment results will be described. Discussions on the efficiency and functionality of the proposed tool will also be included. Finally, conclusion and future development will be discussed in Chapter 5.

Chapter 2

Axial curve-pair based deformation

Axial deformation technique is one of the most popular tools in the world of modeling deformation. Like other freeform deformation techniques, the axial deformation technique requires to manipulate the control points of an axial curve in order to deform an object.

The technique of axial deformation was firstly introduced by Lazarus et al [8]. Peng et al [16] presented an arc length preserving axial deformation technique using a B-spline curve as the axial curve. Based on an arc-length parameterization of the axial curve, homogenous deformation of an object is achieved. Hui [9] presented the axial curve-pair deformation technique [9] which allows an object to be stretched, bended and twisted intuitively by manipulating the curve-pair.

In this section, the idea of axial curve-pair deformation which is the base technique of our work addressed in this thesis will be reviewed. This thesis extends the technique by incorporating physics based deformation in the axial curve-pair deformation.

2.1 Axial deformation technique

The axial deformation allows the shape of a 3D object to be deformed by manipulating an associated axial curve. Given a 3D model S and an axial curve $\mathbf{c}(t)$, the axial deformation technique proposed by Lazarus et al. associates a point $\mathbf{s}$ in S with the curve $\mathbf{c}(t)$. (Figure 2.1)

As a result, $\mathbf{s}$ is defined relative to a local coordinate frame on the curve $\mathbf{c}(t)$. The location of $\mathbf{s}$ will be updated when the location and orientation of the local coordinate frame associated with the curve $\mathbf{c}(t)$ is modified.

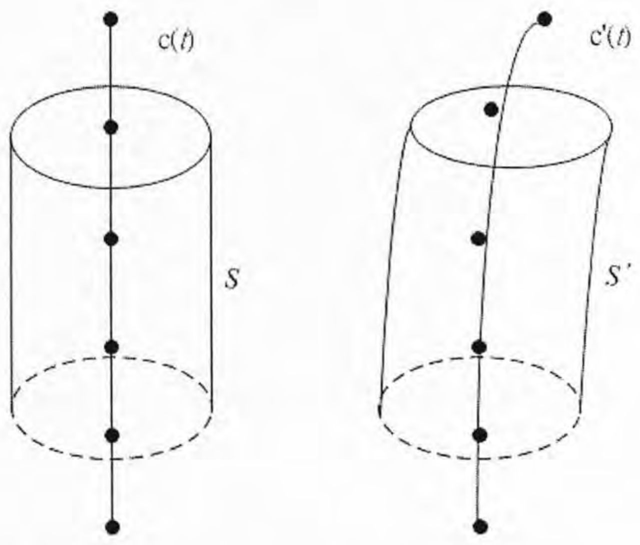


Figure 2.1 - The axial deformation technique

The principal of the technique relies on the representation of an object in the axial space of a curve which will be discussed in the following sections.

2.1.1 Representing objects in axial space

An axial space $A_c\{\mathbf{c}(t), \mathbf{l}(t)\}$ defined by a curve $\mathrm{c}(t)$, $t_{start} \leq t \leq t_{end}$, and a local coordinate frame $\mathbf{l}(t) = [\mathbf{l}_x(t), \mathbf{l}_y(t), \mathbf{l}_z(t)]$ on the curve is a subset of R^4.

A point $\mathbf{s} \in A_c\{\mathbf{c}(t), \mathbf{l}(t)\}$ is defined as (t, u, v, w) where (u, v, w) is the coordinates of the point relative to the local coordinate frame $\mathbf{l}(t)$ at $\mathbf{c}(t)$. A point $\mathbf{s}$ is then defined in the global coordinate frame as $\mathbf{c}(t) + u\mathbf{l}_x(t) + v\mathbf{l}_y(t) + w\mathbf{l}_z(t)$.

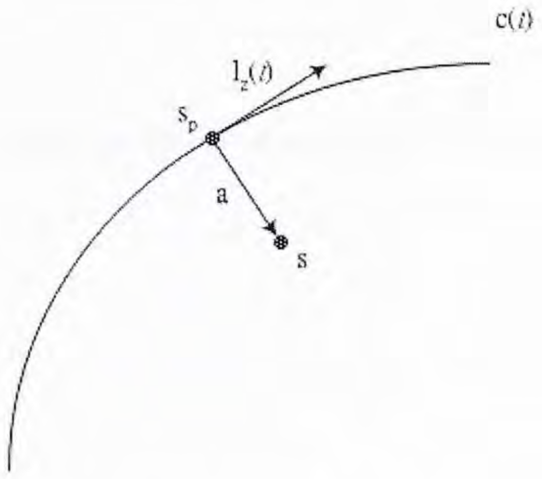


Figure 2.2 - The axial space of axial deformation

To convert a point $\mathbf{s}$ in the global coordinate frame to the axial space of $A_c\{\mathbf{c}(t), \mathbf{l}(t)\}$, the parameter t is taken to be the parametric value of the point $\mathbf{s}_p = \mathbf{c}(t)$ such that $\mathbf{s}_p$ is closest to $\mathbf{s}$ and $\mathbf{l}_z(t)$ represents the direction of the tangent at $\mathbf{c}(t)$. In other words, $\mathbf{s}_p$ is the normal projection of $\mathbf{s}$ onto $\mathbf{c}(t)$.

Refer to Figure 2.2, suppose $\mathbf{a} = \mathbf{s} - \mathbf{s}_p$, $\mathbf{s}$ can be denoted as:

$$\mathbf{s} = \mathbf{c}(t) + [\mathbf{a}\cdot\mathbf{l}_x(t)]\,\mathbf{l}_x(t) + [\mathbf{a}\cdot\mathbf{l}_y(t)]\,\mathbf{l}_y(t) + [\mathbf{a}\cdot\mathbf{l}_z(t)]\,\mathbf{l}_z(t) \qquad \ldots\ldots\ldots\ldots \quad (2.1)$$

The point $\mathbf{s}$ represented in the axial space of $A_c\{\mathbf{c}(t), \mathbf{l}(t)\}$ is then expressed as

$$\mathbf{s} : \quad \begin{cases} (t_{start}, u, v, w),\ \mathbf{s}_p = \mathbf{c}(t_{start}); & t < t_{start} \\ (t, u, v, 0),\ \mathbf{s}_p = \mathbf{c}(t); & t_{start} \le t \le t_{end} \\ (t_{end}, u, v, w),\ \mathbf{s}_p = \mathbf{c}(t_{end}); & t > t_{end} \end{cases} \qquad \ldots\ldots\ldots\ldots \quad (2.2)$$

For $t_{start} \le t \le t_{end}$, $\mathbf{s}_p$ is the normal projection of $\mathbf{s}$ onto $\mathbf{c}(t)$ so that $\mathbf{a} \cdot \mathbf{l}_z(t) = 0$ and hence $\mathbf{s} = (t, u, v, 0)$.

2.1.2 Defining the coordinate frame

Frenet Frame [17] of a curve is commonly used to specify the local coordinate frame of a curve. The Frenet Frame of a curve is a coordinate frame defined in terms of the first and second derivative of the curve. Given a curve $\mathbf{c}(t)$, the Frenet Frame of the curve is defined by the tangent $\mathbf{T}(t)$, normal $\mathbf{N}(t)$ and binormal $\mathbf{B}(t)$ of $\mathbf{c}(t)$. They are expressed as:

$$\mathbf{T}(t) = \mathbf{c}'(t) \,/\, \| \mathbf{c}'(t) \|$$

$$\mathbf{N}(t) = \mathbf{T}'(t) \,/\, \| \mathbf{T}'(t) \|$$

$$\mathbf{B}(t) = \mathbf{T}(t) \times \mathbf{N}(t) \qquad \ldots\ldots\ldots\ldots \quad (2.3)$$

where $\mathbf{c}'(t)$ is the first derivative of $\mathbf{c}(t)$, and $\mathbf{T}'(t)$ is the first derivative of $\mathbf{T}(t)$.

However, as the local coordinate frame is completely defined by the curve $\mathbf{c}(t)$, the orientation of the frame is uncontrollable. Thus, unexpected twist may be obtained. Moreover, the $\mathbf{c}''(t)$ may vanish and hence the coordinate frame could not be defined.

Klok [18] introduced another approach using a rotation-minimizing frame defined with a set of differential equations. Lossing and Eshleman [24] introduced the directional curve approach. Both methods could not provide a solution to fully control the twist of the curve. Hui [9] later introduced the axial curve-pair deformation technique which allows the local coordinate frame to be controlled intuitively.

2.2 Axial curve-pair deformation technique

Hui [9] proposed an intuitive approach to manipulating deformable shape by adopting the axial curve-pair deformation technique. A curve-pair is used as the basic deformation element in order to provide a complete control on the twist of a component.

The use of curve-pair allows the local coordinate frame of an axial curve to be controlled intuitively. By associating an object to the curve-pair, the object can be stretched, bended and twisted intuitively through adjusting the control points of the curve-pair (Figure 2.3).

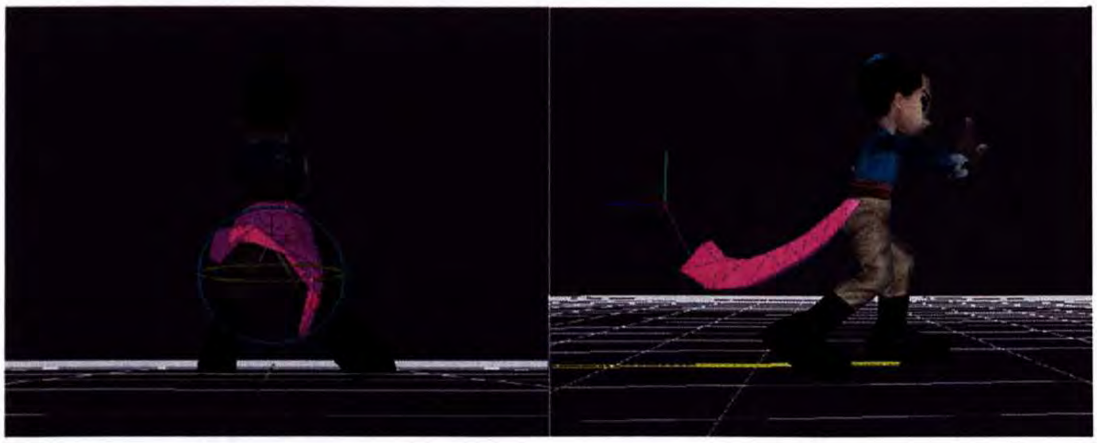


Figure 2.3 - A twisted object and stretched object using the axial curve-pair deformation technique

2.2.1 Framing the curve-pair

A curve-pair is composed of a primary curve $\mathbf{c}(t)$ and an orientation curve $\mathbf{c_D}(t')$, where $t'_{start} \leq t' \leq t'_{end}$. When a vertex of a model $\mathbf{s}$ is attached to the curve $\mathbf{c}(t)$, a projection point $\mathbf{s_p} = \mathbf{c}(t)$ is obtained.

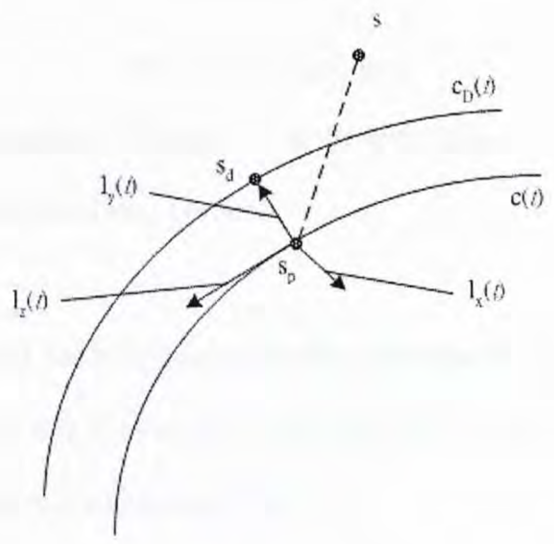


Figure 2.4 - The local coordinate frame of the axial curve-pair

In order to define the position and orientation of the local coordinate frame, a point $\mathbf{s_d}$ is obtained by projecting the point $\mathbf{c_D}(t)$ onto the plane through the projection point $\mathbf{s_p}$ with unit normal $\mathbf{c}'(\mathrm{t}) / |\mathbf{c}'(t)|$ (Figure 2.4). Hence, $(\mathbf{s_p} - \mathbf{s_d}) \cdot \mathbf{c}'(t) = 0$ and the local coordinate frame is given by

$$\begin{aligned} \mathbf{l_z}(t) &= \mathbf{c}'(t) / |\mathbf{c}'(t)| \\ \mathbf{l_x}(t) &= \mathbf{n}(t) \times \mathbf{l_z}(t), \\ \mathbf{l_y}(t) &= \mathbf{l_z}(t) \times \mathbf{l_x}(t) \end{aligned} \qquad \text{..........} \quad (2.4)$$

where $\mathbf{n}(t) = \{\mathbf{c_D}(t) - \mathbf{c}(t)\} / |\mathbf{c_D}(t) - \mathbf{c}(t)|$

2.2.2 Construction of orientation curve

The axial space is defined by the primary curve $\mathbf{c}(t)$ and the orientation curve $\mathbf{c}_D(t)$ as discussed in the previous section. Assume the primary curve is constructed as a B-spline curve and is defined by

$$\mathbf{c}(t) = \textstyle\sum^{n}_{i=0} N_{i,p}(t)\, \mathbf{p}_i \qquad \dots\dots\dots\dots \quad (2.5)$$

where $\mathbf{p}_i$ are the control points of the curve, i is the total number of the control points, and $N_{i,p}(t)$ is the B-spline basis function of order p.

The orientation curve $\mathbf{c}_D(t)$ lies within a circular tube of radius r with $\mathbf{c}(t)$ being the axis of the tube. The offset distance is not the most important issue but the offset direction is essential for defining the local coordinate frame.

The orientation curve is an offset curve of the primary curve. The primary curve $\mathbf{c}(t)$ is first constructed with all the control points lying on a straight line. The orientation curve $\mathbf{c}_D(t)$ is then constructed by creating a planar offset of the primary curve (Figure 2.5). Thus the orientation curve could be defined as

$$\mathbf{c}_D(t) = \textstyle\sum^{n}_{i=0} N_{i,p}(t)\, \mathbf{q}_i \qquad \dots\dots\dots\dots \quad (2.6)$$

where $\mathbf{q}_i = \mathbf{p}_i + d$, d is the offset distance.

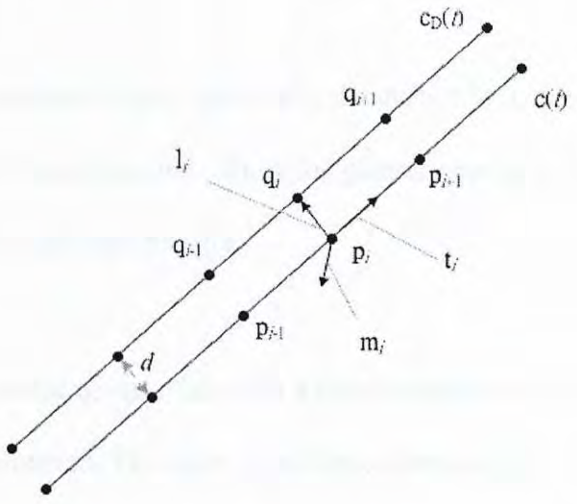


Figure 2.5 - The primary curve and the orientation curve

As a result, an orientation curve $\mathbf{c}_\mathbf{D}(t)$ lies within a circular tube of radius d with $\mathbf{c}(t)$ being the axis of the tube is constructed.

In order to allow the orientation curve to be adjusted in accordance to the primary curve, an alternative method is adopted rather than moving the control point of the orientation curve in accordance with that of the primary curve. This will be discussed in the next section.

2.2.3 Manipulation of the axial curve-pair

To control the local coordinate frame intuitively, the curves in a curve-pair have to be manipulated in a synchronized manner. When the primary curve is modified, the orientation curve has to be modified correspondingly.

The orientation of the vector $\mathbf{q}_i - \mathbf{p}_i$ relative to a local coordinate frame at $\mathbf{p}_i$ is kept consistent while $\mathbf{p}_i$ is relocated. The local coordinate frame at $\mathbf{p}_i$ is specified with a polygon tangent at $\mathbf{p}_i$ and a vector normal to the polygon tangent.

There are a number of ways for defining the tangent at a control point. The method adopted in the work is defined as followings:

Given a polygon with vertices $\mathbf{p}_\mathrm{i}$, then polygon tangent $\mathbf{t}_\mathrm{i}$ at $\mathbf{p}_\mathrm{i}$ is given by

$$\mathbf{t}_i = \begin{cases} (\mathbf{p}_{i+1} - \mathbf{p}_i) / |\mathbf{p}_{i+1} - \mathbf{p}_i|, & i = 0 \\ (\mathbf{p}_{i+1} - \mathbf{p}_i) / |\mathbf{p}_{i+1} - \mathbf{p}_i| + (\mathbf{p}_i - \mathbf{p}_{i-1}) / |\mathbf{p}_i - \mathbf{p}_{i-1}|, & 0 < i < n \\ (\mathbf{p}_i - \mathbf{p}_{i-1}) / |\mathbf{p}_i - \mathbf{p}_{i-1}|, & i = n \end{cases} \qquad \dots\dots\dots\dots (2.7)$$

Given an axial curve-pair $(\mathbf{c}, \mathbf{c}_\mathbf{D})$, the local coordinate frame at $\mathbf{p}_i$ is given by unit vectors $\mathbf{l}_i$, $\mathbf{m}_i$, $\mathbf{t}_i$, where $\mathbf{t}_i$ is the polygon tangent at $\mathbf{p}_i$, and $\mathbf{l}_i = (\mathbf{q}_i - \mathbf{p}_i) \times \mathbf{t}_i \, |\mathbf{q}_i - \mathbf{p}_i|$, $\mathbf{m}_i = \mathbf{t}_i \times \mathbf{l}_i$ (Figure 2.6).

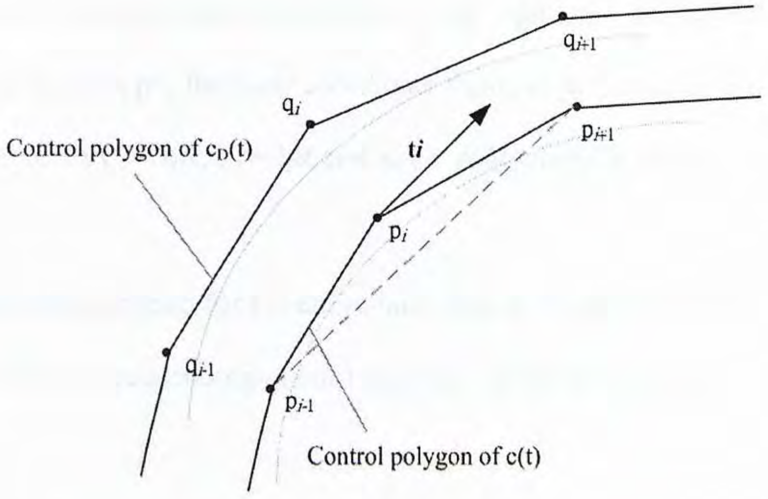


Figure 2.6 - The control polygon and polygon tangent

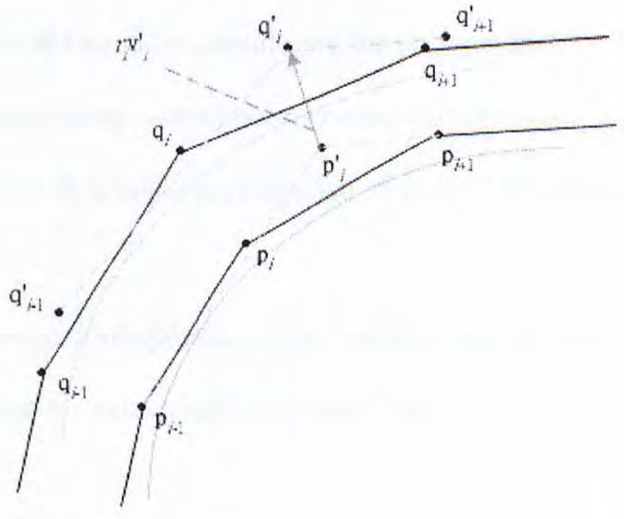


Figure 2.7 - Moving a control point of the primary curve

Changing one control point will affect the orientation of the polygon tangents at the preceding and succeeding control points. The neighboring control points of the modified control point of the orientation curve have to be adjusted in order to maintain a consistent relationship between the two curves.

Consider the polygon tangent $\mathbf{t}_i$ and the vector $\mathbf{v}_i = (\mathbf{q}_i - \mathbf{p}_i) / |\mathbf{q}_i - \mathbf{p}_i|$ in Figure2.6 and 2.7, by moving $\mathbf{p}_i$ to a location $\mathbf{p'}_i$, the local coordinate frame at $\mathbf{p}_i$ is rotated and the polygon tangent $\mathbf{t}_i$ at $\mathbf{p}_i$ becomes $\mathbf{t'}_i = \mathbf{t}_i\mathbf{R}$, $\mathbf{l'}_i = \mathbf{l}_i\mathbf{R}$ and $\mathbf{m'}_i = \mathbf{m}_i\mathbf{R}$ where $\mathbf{R}$ is the rotation matrix.

Assuming a fixed configuration for the curve-pair, then $\mathbf{v}_i = a_i\mathbf{m}_i + b_i\mathbf{t}_i$ gives $\mathbf{v'}_i = a_i\mathbf{m'}_i + b_i\mathbf{t'}_i$. Hence, $\mathbf{v'}_i = \mathbf{v}_i\mathbf{R}$. The corresponding control point $\mathbf{q'}_i$ on the orientation curve can be obtained by

$$\mathbf{q'}_i = r_i\mathbf{v'}_i + \mathbf{p'}_i. \qquad \text{..........} \quad (2.8)$$

The rotation matrix $\mathbf{R}$ is obtained by considering the polygon tangent $\mathbf{t'}_i$ at $\mathbf{p'}_i$. Since the distance between corresponding control points remains unchanged,
$\mathbf{r} = (\mathbf{t}_i \times \mathbf{t'}_i) / |\mathbf{t}_i \times \mathbf{t'}_i|$ which is the axis of rotation of $\mathbf{t}_i$ could be obtained when $\mathbf{p}_i$ is moved.

Since the distance between corresponding control points remains unchanged, and the vector $\mathbf{v}_i$ rotates about $\mathbf{r}$ through the same angle as $\mathbf{t}_i$ rotates about $\mathbf{r}$, thus,

$$\mathbf{v}_i \times \mathbf{v'}_i = \mathbf{t}_i \times \mathbf{t'}_i. \qquad \text{..........} \quad (2.9)$$

Hence, the vector $\mathbf{v'}_i$ and the control point $\mathbf{q'}_i$ of the orientation curve can be obtained from the Eq. (2.9). The control points $\mathbf{q'}_{i-1}$ and $\mathbf{q'}_{i+1}$ are evaluated with the same method.

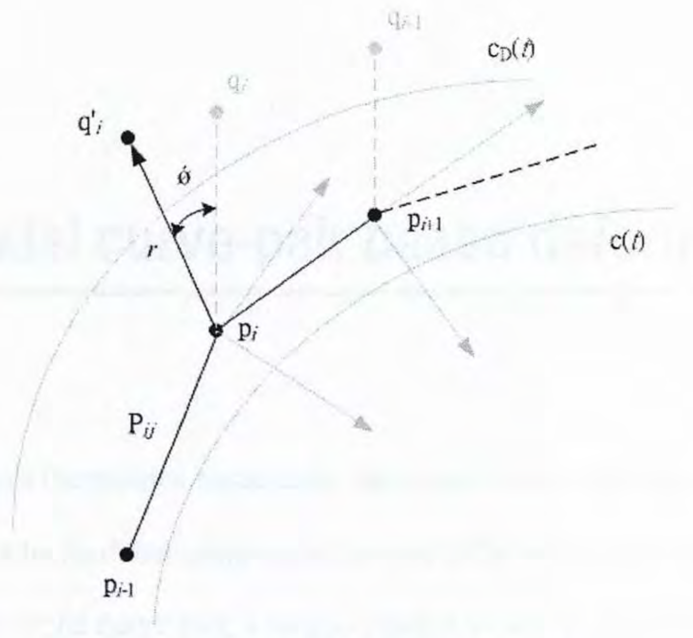


Figure 2.8 - Adjusting the orientation curve

Twisting deformations of an object are achieved by twisting the orientation curve around the primary curve (Figure 2.8). Hence, a twist can be performed by rotating $\mathbf{q}_i$ around the polygon tangent $\mathbf{t}_i$ at $\mathbf{p}_i$ such that $\mathbf{q}_i$ lies inside a sphere of radius r centered at $\mathbf{p}_i$.

Since the polygon tangent $\mathbf{t}_i$ at $\mathbf{p}_i$ is fixed for a given primary curve, thus, only rotation of the frame about $\mathbf{t}_i$ can be modified. Using the relation $\mathbf{q}_i = r_i\,(\,a_i\mathbf{m}_i+b_i\mathbf{t}_i\,)$, a rotation of $(\mathbf{l}_i, \mathbf{m}_i, \mathbf{t}_i)$ about $\mathbf{t}_i$ is the same as a rotation of $\mathbf{q}_i$ about $\mathbf{t}_i$.

Chapter 3

Dynamic axial curve-pair based deformation

This chapter formulates the physics-based axial curve-pair model. The dynamics of a deformable object can be modeled using one of several different methods. Because of the unique representation of the curve-pair, a special physics-based formulation is adopted.

The formulations of dynamic NURBS [10] and inter-lattice spring mass system [4][5][11][12][13][14][15] cannot precisely model the physics-based axial curve-pair because their dynamic configurations are applied to the lattices directly. In the axial curve-pair formulation, the orientation curve is deformed automatically corresponding to the primary curve. The twisting deformation is modeled by rotating the orientation curve about the primary curve where the physical properties are expressed by the angles rotating about the tangent of a curve instead of the lattice points themselves.

In our approach, an explicit formulation, the mass spring system, is used and the deformation is applied directly on a flexible model, the curve-pair. This is achieved by fitting the curve-pair to the mass-spring system. The mass-spring system gives an effective and satisfactory estimation of object deformation. By implementing tensile interactions between the neighboring point masses in a curve-pair mass spring structure, the mechanical behaviors of the curve-pair are usually limited to linear elasticity only. Deformation such as bending and twisting can not be simulated smoothly.

A special mass-spring model is deigned for the curve-pair representation which allows bending and twisting actions of the curve-pair. To model different elastic behaviors of the curve-pair with shape restoring properties, each point mass is linked to its neighbors by different types of springs. The springs tend to keep the point masses at their initial resting positions. To model the twisting properties of the curve-pair, each coordinate frame at each point mass is linked to its neighboring coordinate frames by torsional springs. The dynamics are modeled by updating the mass spring system at discrete time-steps. In each step, the spring forces are calculated and applied to the point masses, which respond by accelerating in the direction of the net force.

The forces exerted on each point mass depend on the current state of the system, which is determined by the location of the point mass, orientation of the local coordinate frames and the external interactions. These forces include forces due to internal elasticity and viscosity forces, gravity and wind force, etc.

Using a forth order Runge-Kutta method with adaptive step-size, the equations of motion are integrated and the position and velocity of each point mass can be obtained. The curve-pair is fitted to the deformed positions of the mass spring system which determines the resulting object deformations.

3.1 Overview on physic-based formulations

There are a number of formulations to model the physical behaviors of a deformation tool. Deformation techniques like axial curve deformation and axial curve-pair deformation allow users to deform an object by manipulating the control points of a curve or curve-pair. The control polygon of an axial curve-pair is shown in Figure 3.1.

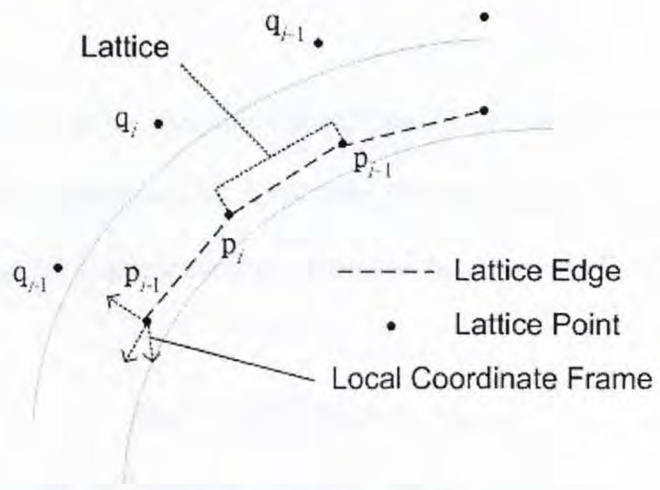


Figure 3.1 - The lattices of the axial curve-pair model

An axial curve-pair bending deformation is mainly controlled by the primary curve. The twisting deformation is handled by the orientation curve in a special manner. The twisting motion is modeled by rotating the orientation curve about the primary curve instead of the deformation of lattice points.

Dynamic NURBS [10] curve and the Dynamic FFD [6] model a physical deformation by applying a dynamic formulation to the lattices. These formulations will form the basis of a dynamic axial curve-pair model.

3.1.1 Dynamic NURBS Curve

Dynamic NURBS curve [10] proposed a physics-based formulation which extends the geometric NURBS curve definition by explicitly incorporating a time variable.

$$\mathbf{c}(u, t) = \sum^{n}_{i=0} B_{i,p}(u)\, \mathbf{p}_i(t)\, \mathrm{w}_i(t) \,/\, \sum^{n}_{i=0} B_{i,p}(u)\, \mathrm{w}_i(t) \qquad \ldots\ldots\ldots\ (3.1)$$

where the $B_{i,p}(u)$ are the usual recursively defined piecewise basis functions [22], $\mathbf{p}_i(t)$ are the $n+1$ control points and $\mathrm{w}_i(t)$ are associated nonnegative weights.

The equations of motion of the dynamic NURBS model are derived from the work-energy version of Lagrangian dynamics [23]. A second-order non-linear equation of motion is obtained by applying the Lagrangian formulation to the dynamic NURBS model:

$$\mathbf{Mp}'' + \mathbf{Dp}' + \mathbf{Kp} = \mathbf{f}_p + \mathbf{g}_p \qquad \ldots\ldots\ldots\ (3.2)$$

where the mass matrix **M**, the damping matrix **D** and the stiffness matrix **K** can be formulated explicitly [3][10][24].

Since the deformation of the axial curve-pair is defined in both lattice points and lattice edges (Figure 3.1), the dynamic NURBS model which is applied to the geometric NURBS curve could not be applied to the axial curve-pair directly. The dynamic NURBS model could only model the stretching of the curve-pair which involves the lattice points only. The bending of the curve-pair requires formulating the relationship between the lattice points and the local coordinate frame at lattice point in order to define a solid representation of the original shape of the curve-pair. We will have a further discussion in Section 3.1.3.

On the other hand, we need to define the relationship between the dynamics of the two curves in the curve-pair model. The twisting of the dynamic curve-pair requires modeling a correlation between the local coordinate frames at two successive lattices. These simply cannot be modeled by D-NURBS as the bending and twisting of the curve-pair involves local coordinate frames at different lattices.

3.1.2 Dynamic Free Form Deformations

The Free Form Deformation technique [1] which consists of embedding the geometric model into a user-defined 3D lattice space is defined in terms of a tensor product Bernstein polynomial. A local 3D coordinate system is imposed on a parallelepiped with the three basis vectors **u**, **v** and **w** forming the axes of the system. Dynamic Free Form Deformation [5] extends the formulation similar to the D-NURBS by explicitly incorporating time.

A general dynamic formulation of Free Form Deformation is:

$$\mathbf{p}(u,v,w,t) = \sum^{l}_{i=0} \sum^{m}_{j=0} \sum^{n}_{k=0} B_{i,l}(u)\, B_{j,m}(v)\, B_{k,n}(w) \mathbf{K}_{i,j,k}(t) \quad \dots\dots\dots \quad (3.3)$$

where t is the incorporating time, **K** is the position of the control points, and B is the Bernstein polynomial.

The dynamics configuration is defined by embedding the control points in a mass-spring system which is constructed by linear springs. The physical behavior of the control points will be expressed on the deformation of embedded object.

Although a mass-spring system is effective in expressing the physical properties, the properties could not be reflected on the embedded objects since the dynamic configuration is applied to the control points of the deformation lattice.

3.1.3 Dynamic axial curve-pair deformation

Using the same formulation applied to the lattices of an axial curve-pair, a physics-based axial curve deformation can be modeled. In an axial curve-pair, the orientation curve will be deformed automatically corresponding to the primary curve. However, there are some drawbacks about applying such formulation to the axial curve-pair model:

A) The axial curve deformation cannot fully reflect the physical behavior of lattices deformation.

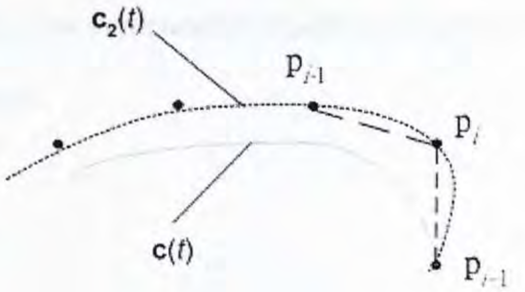


Figure 3.2 - The bending deformation of an axial curve model

The lattice deformation applied to the control points means the dynamic configuration cannot be applied to the curve directly. Figure 3.2 shows the a bending and stretching deformation of a curve **c**(t). The lattices is bended by nearly 90 degrees but the deformed curve **c**(t) does not show the actual bending and stretching power. On the other hand, the dotted curve $\mathbf{c}_2(t)$ which is fitted to the control points reflects the bending and stretching power with its length and shape approximated by the deformed lattices directly.

B) The twisting deformation of the axial curve pair is not modeled in the dynamic formulation.

Since the twisting deformation of the axial curve-pair model is controlled by the orientation curve, the standard control lattices of the curve cannot be used to control the twist of a curve pair. To model the elastic twisting behavior, a dynamic configuration is required to be designed specially for the axial curve-pair. The deformation energy stored in each local coordinate frame can be expressed in a mathematical form and further presented by rotating the orientation curve.

C) The dynamic axial curve deformation based on simple lattices does not have shape restoring power.

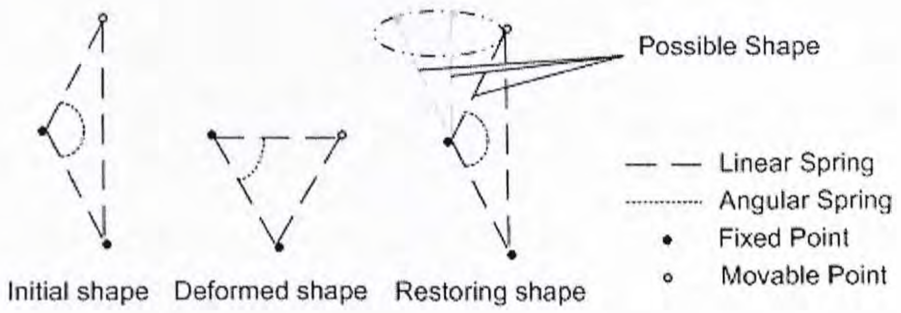


Figure 3.3 - The restoring deformation of a simple mass-spring system

Embedding the lattices in a mass-spring system constructed by linear and angular springs does not guarantee the original shape to be restored after a deformation. Figure 3.3 shows different configurations of a mass-spring system in different processes. An angular springs connecting neighboring lattice edge or bending springs connecting the lattice points can result in an unpredictable shape.

Concluding the pros and cons of the dynamic formulations used in different deformation tools, none of them can be directly applied to the dynamic axial curve-pair model. However, they are useful for forming the basis of dynamic axial curve-pair formulation. The mass-spring system is chosen because of its highly efficient emulation power and low computational complexity.

The primary curve of an axial curve-pair could be modeled by embedding a curve in a mass-spring system. A curve fitted to the mass points of the system can also help to emulate the physical behavior more accurately. Special springs can be designed for the mass-spring system in order to emulate the twisting deformation and improve the shape restoring power. The following sections will describe the proposed dynamic axial curve-pair model in detail.

3.2 The dynamic mass-spring model

Denote the primary curve as $\mathbf{c}(t)$, and the orientation curve as $\mathbf{c}_\mathbf{D}(t)$, the curve-pair can be constructed as a pair of k th-degree non-rational B-spline curves:

$$\mathbf{c}(t) = \sum\nolimits^{n}_{i=0} N_{i,p}(t)\ \mathbf{u}_i$$

$$\mathbf{c}_\mathbf{D}(t) = \sum\nolimits^{n}_{i=0} N_{i,p}(t)\ \mathbf{v}_i \qquad \dots\dots\dots\dots \quad (3.4)$$

where $\mathbf{u}_i$ is the i-th control point of the primary curve, $\mathbf{v}_i$ is the i-th control point of the orientation curve, $N_{i,p}(t)$ is the B-spline basis function of degree p.

In order to emulate the best deformation properties of the mass spring system, the curve-pair is constructed to pass through a set of point masses of the dynamic system.

3.2.1 Curve fitting problem

Suppose the primary curve $\mathbf{c}(t)$ interpolates point masses $\mathbf{p}_k$ and the orientation curve $\mathbf{c}_\mathbf{D}(t)$ interpolates point masses $\mathbf{q}_k$. If we assign a parameter value r_k to each sampling points $\mathbf{p}_k$ and $\mathbf{q}_k$, and select an appropriate knot vector $K = \{k_0, \dots, k_m\}$, a $(n+1)$ x $(n+1)$ system of linear equations can be obtained.

$$\mathbf{p}_k = \mathbf{c}(r_k) = \sum\nolimits^{n}_{i=0} N_{i,p}(r_k)\ \mathbf{u}_i$$

$$\mathbf{q}_k = \mathbf{c}_\mathbf{D}(r_k) = \sum\nolimits^{n}_{i=0} N_{i,p}(r_k)\ \mathbf{v}_i \qquad \dots\dots\dots\dots \quad (3.5)$$

where the control points $\mathbf{u}_i$ and $\mathbf{v}_i$ are the $n+1$ unknowns with dimension three.

Because there are n+1 B-spline basis functions and n+1 parameters r_k, this gives a (n+1) x (n+1) matrix **N**.

$$\mathbf{N} = \begin{bmatrix} N_{0,p}(r_0) & \cdots & N_{n,p}(r_0) \\ \vdots & \cdots & \vdots \\ N_{0,p}(r_n) & \cdots & N_{n,p}(r_n) \end{bmatrix} \quad \ldots\ldots\ldots\ldots \quad (3.6)$$

Combining Eq. (3.5) and Eq. (3.6), and expressing in matrix form gives:

$$\mathbf{P} = \mathbf{N} \cdot \mathbf{U}$$
$$\mathbf{Q} = \mathbf{N} \cdot \mathbf{V} \quad \ldots\ldots\ldots\ldots \quad (3.7)$$

where **P**, **Q**, **U** and **V** are (n+1) x 3 matrices.

Since **P** and **Q** are input data and **N** can be obtained by evaluating the B-spline basis functions at the given parameters, therefore **U** and **V** are the unknowns to be determined. The system can be solved by using Gaussian elimination without pivoting and with knots computed by averaging consecutive k parameters.

The choice of r_k and K affects the shape and parameterization of the curve. A curve passing through the set of mass points and which interpolate along the structural spring skeleton is preferred. Popular approaches for choosing r_k are the uniform method, chord length distribution, centripetal method and the universal method. It is found that the centripetal parameter selection method with knots evaluated by the technique of averaging gives the best result especially when the data points are not evenly distributed, and there are sharp turns in the curve.

Let e be the sum of the square root of chord length

$$e = \sum^{n}_{k=0} | \mathbf{p}_k - \mathbf{p}_{k-1} |^{1/2} \qquad \text{..........} \quad (3.8)$$

Then, we have

$$r_0 = 0 \text{ and } r_n = 1$$

$$r_k = r_{k-1} + | \mathbf{p}_k - \mathbf{p}_{k-1} |^{1/2} / e \qquad \text{..........} \quad (3.9)$$

where $k = 1, \ldots, n\text{-}1$

With knots evaluated by the technique of averaging:

$$k_0 = \ldots = k_p = 0,\ k_{m-p} = \ldots = k_m = 1$$

$$k_{j+p} = \sum^{j+p-1}_{i=j} r_i \qquad \text{.........} \quad (3.10)$$

where m is the number of knots, $j = 1, \ldots, n - p$.

Combining Eq. (3.9) and Eq. (3.10) to evaluate r_k will lead to a system which is positive and banded with a semi-bandwidth less than p.

3.2.2 Construction of the dynamic curve-pair

The primary curve **c**(t) passes through point masses $\mathbf{p}_i$ and the orientation curve $\mathbf{c}_\mathbf{D}(t)$ passes through point masses $\mathbf{q}_i$ of the mass spring system. The orientation curve is constructed such that $\mathbf{q}_i = \mathbf{p}_i + d$, where d is the offset distance (Figure 3.4).

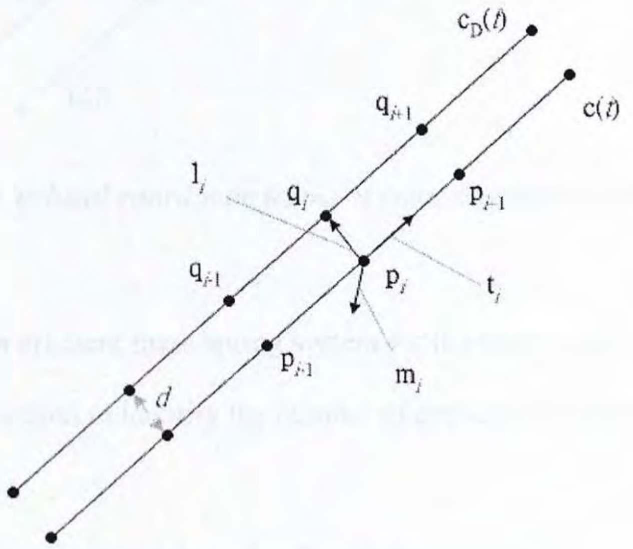


Figure 3.4 - The axial curve-pair model

Given an axial curve-pair (**c**, $\mathbf{c}_\mathbf{D}$), the local coordinate frame at $\mathbf{p}_i$ is given by unit vectors $\mathbf{l}_i$, $\mathbf{m}_i$, $\mathbf{t}_i$, where $\mathbf{t}_i$ is the polygon tangent at $\mathbf{p}_i$, and $\mathbf{l}_i = (\mathbf{q}_i - \mathbf{p}_i) \text{ x } \mathbf{t}_i \,|\mathbf{q}_i - \mathbf{p}_i\,|$, $\mathbf{m}_i = \mathbf{t}_i \text{ x } \mathbf{l}_i$. Techniques for manipulating the curve-pair is the same as that described in Chapter 2.

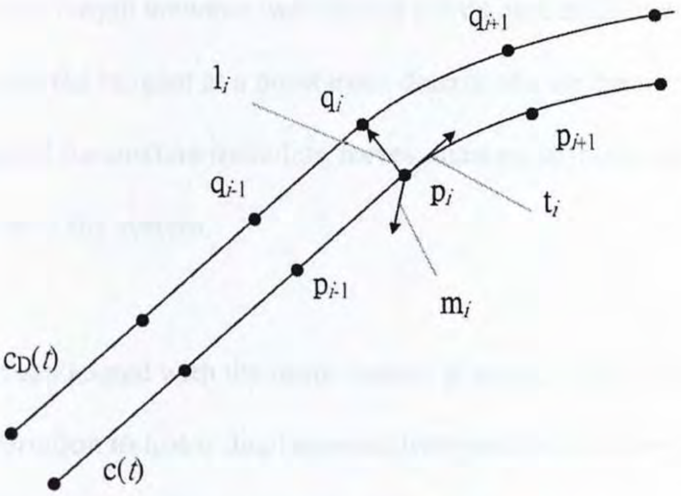


Figure 3.5 - The local coordinate frame at control points of primary curve

Before constructing an efficient mass spring system for the special curve-pair representation, it is essential to identify the number of degrees of freedom of the curve-pair.

Refer to the Figure 3.5, the point mass $\mathbf{p}_i$ has five degrees of freedom including translations in the directions $\mathbf{t}_i$, $\mathbf{l}_i$ and $\mathbf{m}_i$ and rotation about the axis $\mathbf{l}_i$, and $\mathbf{m}_i$. The point mass $\mathbf{q}_i$ has six degrees of freedom including translations in the directions $\mathbf{t}_i$, $\mathbf{l}_i$ and $\mathbf{m}_i$ and rotation about the axis $\mathbf{t}_i$, $\mathbf{l}_i$, and $\mathbf{m}_i$.

The motion of $\mathbf{p}_i$ is governed by the mass-spring system to simulate the physical bending and stretching motion of the curve-pair. However, the motion of $\mathbf{q}_i$ is not fully controlled by the system. When $\mathbf{p}_i$ is modified, $\mathbf{q}_i$ has to be modified correspondingly and automatically. Only the rotation about $\mathbf{t}_i$ of $\mathbf{q}_i$ is governed by the dynamic system to simulate the physical twisting properties of curve-pair.

Next the geometric parameters and the physical parameters are identified. Geometric parameters such as rest length between two control points, rest angle between two local coordinate frames, and the tangent at a point mass determines the geometry of the mass spring system. Physical parameters including forces, masses, stiffness and stress governs the physical behavior of the system.

A mass density **m** is associated with the point masses $\mathbf{p}_i$ and $\mathbf{q}_i$. The linear stress at a point mass is due to the variation in linear displacement from rest-length along its connected spring elements. The torsional stress at a point mass is due to the variation in angular displacement from rest-angle between the lattice edge and the local coordinate frame at lattice point (Figure3.1). To model these stress behavior, special spring elements are connected between the point masses. Different spring elements are adopted to model the different physical behavior of a curve-pair.

Two types of springs are adopted. They are the metric springs and the torsional springs (Figure1.1). The structure of the mass-spring model is specially designed for the curve-pair in order to maintain its unique structure over a set of defined parameters.

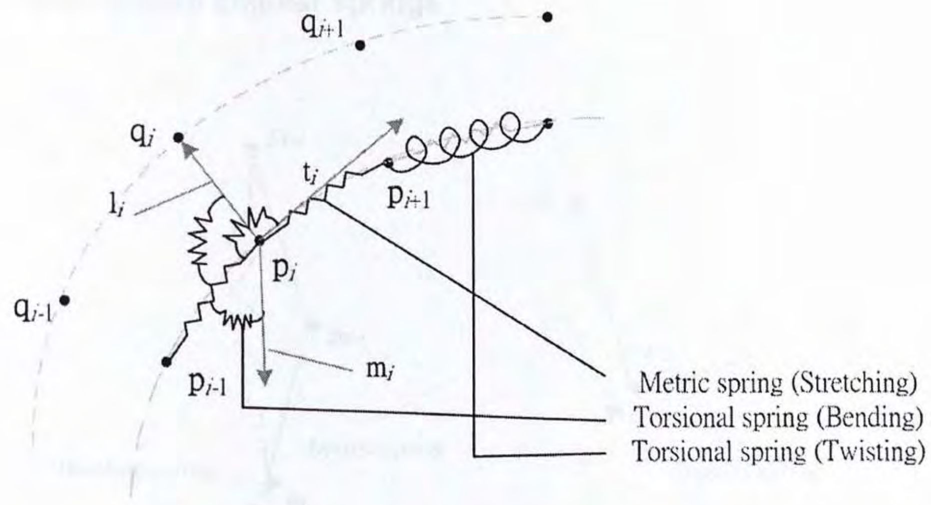

q$_{i+1}$
q$_i$
l$_i$
t$_i$
p$_{i+1}$
p$_i$
q$_{i-1}$
p$_{i-1}$
m$_i$
Metric spring (Stretching)
Torsional spring (Bending)
Torsional spring (Twisting)

Figure 3.6 - The structure of mass-spring model

To maintain the linear structure and to model the tensile stress behavior, metric spring elements are connected between the neighboring point masses ($\mathbf{p}_i \rightarrow \mathbf{p}_{i+1}$) of the mass spring model (Figure 3.6).

Elastic bending is modeled by two types of spring elements. The angular spring elements defined at the angle of the lattice edge and the local coordinate frame axes of the neighboring lattice point provide a controllable bending deformation. To model the torsional stress behavior, torsional spring elements are connected between the local coordinate frames ($\mathbf{l}_i$, $\mathbf{m}_i$, $\mathbf{t}_i$) at $\mathbf{p}_i$ and ($\mathbf{l}_{i+1}$, $\mathbf{m}_{i+1}$, $\mathbf{t}_{i+1}$) at $\mathbf{p}_{i+1}$ (Figure 3.6).

3.2.3 Three-degree angular springs

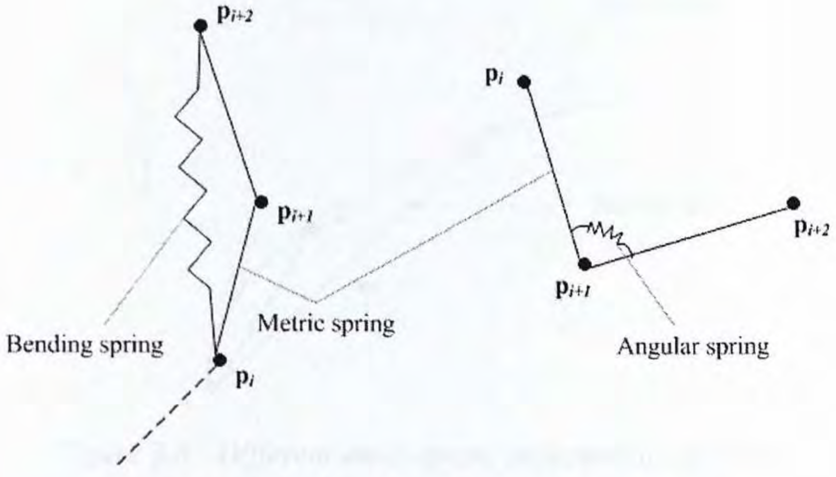


Figure 3.7 - Bending and angular springs

Our mass-spring model is specially designed for the curve-pair representation. In general, to model the bending stresses, bending spring elements are defined between the point masses ($\mathbf{p}_i \rightarrow \mathbf{p}_{i+2}$) or angular spring elements are defined at the angle between neighboring edges.

However, this spring structure for bending deformations will lead to unpredictable results if it is applied to a one dimensional line segment representation. In Figure 3.8, different equilibrium states of the same model are found where w is the rest length of the bending spring,

Bending deformations are derived from the mass-spring system. At the same time, different mass-spring structures are obtained. The same holds when the bending springs are replaced by angular springs shown in Figure 3.7, which is a result of the lack of control on the bending directions.

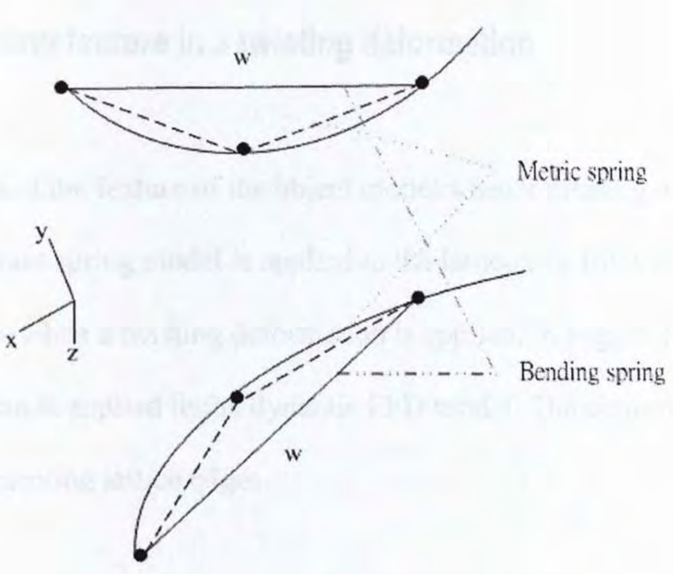


Figure 3.8 - Different mass-spring structures at rest state

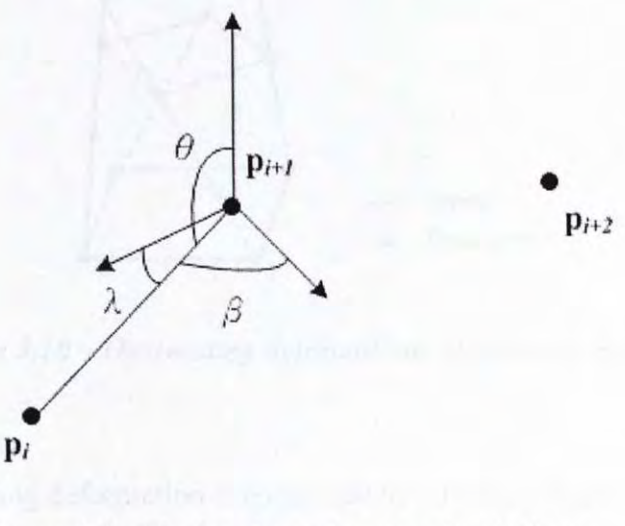


Figure 3.9 - The local angular coordinate frame

In the representation of curve-pair, local coordinate frames at the control points are well defined (Figure3.9). With these local coordinate frames, we can control the bending directions by incorporating an angular spring between the edge and the local coordinate frame axes. As a result, the bending deformations is governed by (λ , θ , β) where ($\cos\lambda$)^2 + ($\cos\theta$)^2 + ($\cos\beta$)^2= 1. This structure ensures predictable and unique bending deformations in most cases.

3.2.4 Conserving feature in a twisting deformation

Our system preserved the feature of the object model when a twisting deformation is applied to it. If a mass spring model is applied to the lattice of a FFD, the object may lose some of its features when a twisting deformation is applied. In Figure 3.10, a 90-degree twisting deformation is applied to the dynamic FFD model. The control lattice collapses as a result of the intersecting lattice edges.

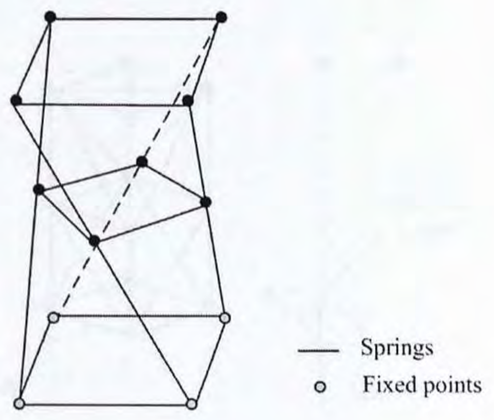


Figure 3.10 - The twisting deformations of dynamic FFD

In our system, the twisting deformation is controlled by a twisting curve where the twisting torque is applied along two neighboring coordinate frames. Therefore the features of the object can be maintained but transformed. There is limitation on the twisting angle which is the angle between two neighboring coordinate frame in our system. In order to obtain a predictable result, the twisting angle cannot exceed 360 degrees and the twisting angle can not exceed 180 degrees in each twisting step. When the twisting angle is greater than the limit, the twisting deformation obtained may not be desirable as the two curves intersect each other. The local coordinate frame at the intersecting point will be corrupted as well as the embedded objects.

3.2.5 Comparison of mass spring models

In [4][5], dynamics are also applied to the deformation computation by incorporating a mass spring system on the FFD control lattices. Compared with these mass spring model, our model uses less point masses and springs in deforming an object. The computational cost of a mass spring system depends on the number of masses and springs. System with dense masses and springs will have higher the computational complexity.

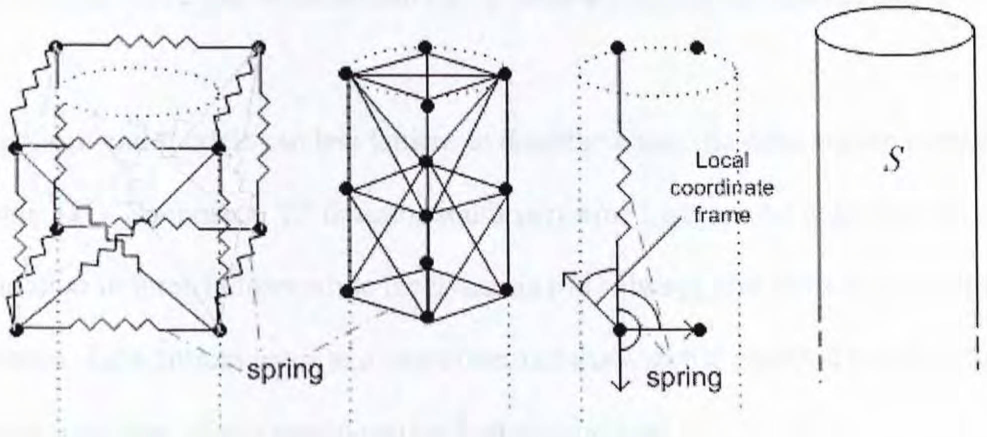


Figure 3.11 - The mass-spring structure of different deformation approaches: (left to right) a dynamic FFD lattice, a spring-mass model based on the geometry of object S, a dynamic axial curve-pair deformation lattice, the deformable object S.

When a mass-spring system is directly applied to the geometry of an object *S*, the number of springs and masses depends on the complexity of the object. If there are 100 vertices in *S*, that is, 100 point masses, then the number of springs in the system may be more than 800. On the other hand, the number of masses and springs used in a dynamic FFD model and dynamic axial curve-pair deformation model is independent of the shape complexity of object *S* (Figure 3.11).

In one lattice, dynamic FFD [4][5][6] requires 8*n* masses and 18*n* springs while dynamic axial curve-pair deformation model requires 2*n* masses and 9*n* springs, where *n* is any positive integer.

The mass-spring structure adopted in the curve-pair model is much simpler when comparing with the mass-spring structure required for a FFD model and the mass-spring structure applied directly on the deformable 3D model. Since computational complexity is proportional to the number of masses and springs used in a mass-spring structure, the dynamic axial curve-pair deformation can achieve a faster computational speed.

Furthermore, our models use less lattices to describe a specific deformation comparing with dynamic FFD. Suppose a "Z" deformation is performed, our model describes the deformation in three lattices while the dynamic FFD always uses more to prevent the lost of features. Less lattices leads to a more compact mass spring model. Therefore the computational cost of our model can be further decreased.

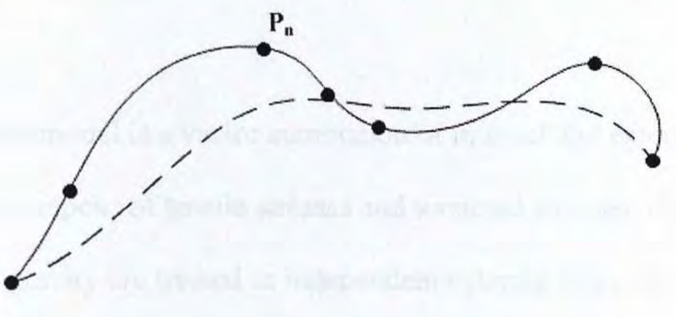


Figure 3.12 - A curve fitted to $\boldsymbol{P_n}$ *(solid line) and a B-spline curve (dashed line) with control points* $\boldsymbol{P_n}$

In the work of [4][5][6], the system dynamics is not directly applied to the core deformable geometry, but is applied to the control points of deformation tool. As a result, deformation of the object based on the core deformation skeleton may not be satisfactory. For example, in Figure 3.12, if dynamics is implied to the control points of the curve-pair, the resulting curve-pair (dashed line) can not fully reflect the deformation properties of the system.

In our approach, the primary axial curve is a cubic B-spline curve fitted to the point masses of the mass spring system. As a result, the fitted curve-pair reflects faithfully the deformation properties of the mass spring system.

3.3 Internal and external forces

The total force on the model is a vector summation of internal and external forces. The internal forces compose of tensile stresses and torsional stresses. User-interactions, reaction forces and gravity are treated as independent external force. In this section, the formulations of different forces are described.

3.3.1 Tensile stress

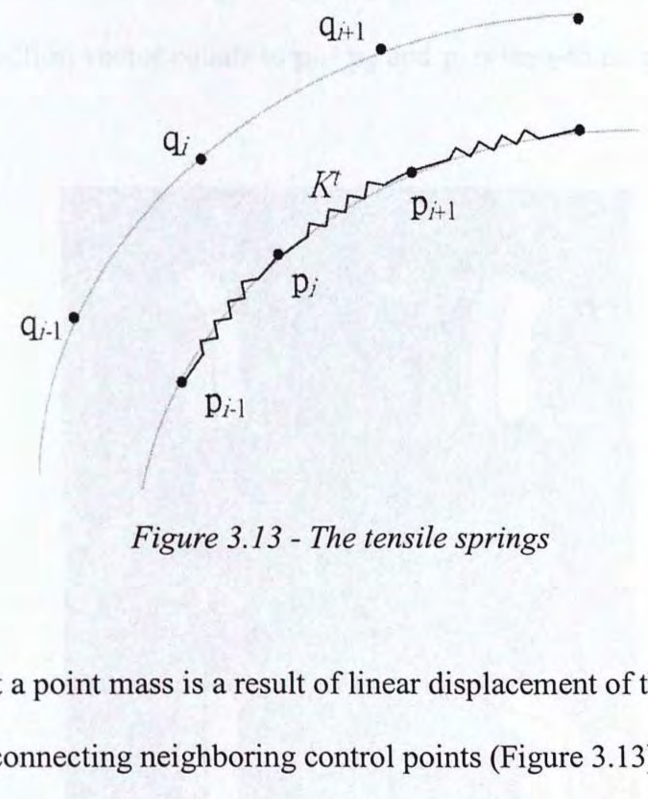


Figure 3.13 - The tensile springs

The tensile stress at a point mass is a result of linear displacement of the point mass along the metric springs connecting neighboring control points (Figure 3.13). The tensile stress at a point mass **p** with spring of stiffness K^t and initial length δ connecting its j-th neighbor is defined as:

$$\mathbf{F}^{\delta}{}_i = K^t{}_i [(\mathbf{p} - \mathbf{p}_j) + \delta_j \mathbf{u}_j] \qquad \ldots\ldots\ldots \quad (3.11)$$

where $\mathbf{u}_j$ is the unit vector which is equal to $(\mathbf{p} - \mathbf{p}_j) / | \mathbf{p} - \mathbf{p}_j |$.

3.3.2 Torsional stress

The torsional stress at a point mass is introduced by angular displacement difference along the angular springs connecting neighboring local coordinate frames. There are four types of angular deformations (Figure 3.14):

i) Bending deformations at $\mathbf{p}_i$ along the angles formed between $\mathbf{P}_{ij}$ and $\mathbf{t}_j$

ii) Bending deformations at $\mathbf{p}_i$ along the angles formed between $\mathbf{P}_{ij}$ and $\mathbf{l}_j$

iii) Bending deformations at $\mathbf{p}_i$ along the angles formed between $\mathbf{P}_{ij}$ and $\mathbf{m}_j$

iv) Twisting deformations on $\mathbf{P}_{ij}$ along the angles formed between $\mathbf{l}_i$ and $\mathbf{l}_j$

where $\mathbf{P}_{ij}$ is the direction vector equals to $\mathbf{p}_i - \mathbf{p}_j$, and $\mathbf{p}_j$ is the j-th neighbor of $\mathbf{p}_i$.

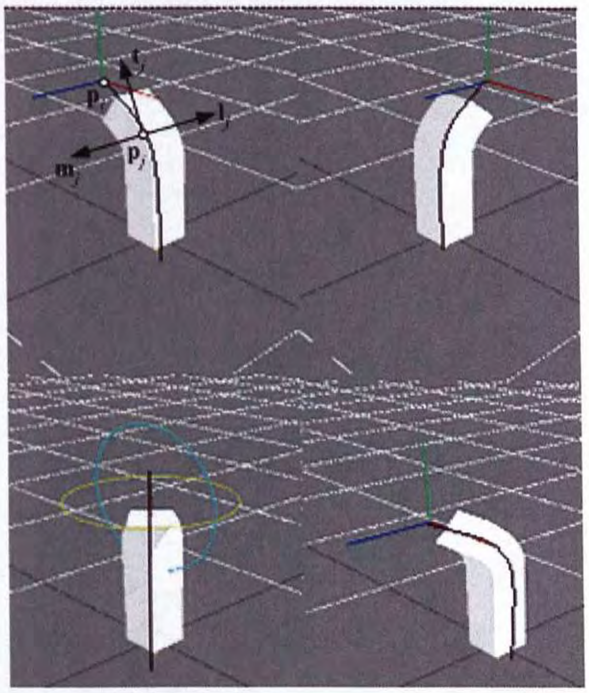


Figure 3.14 - The four types of angular deformations:

Type iii (upper-left), Type ii (upper-right),

Type iv (bottom left) and Type i (bottom-right)

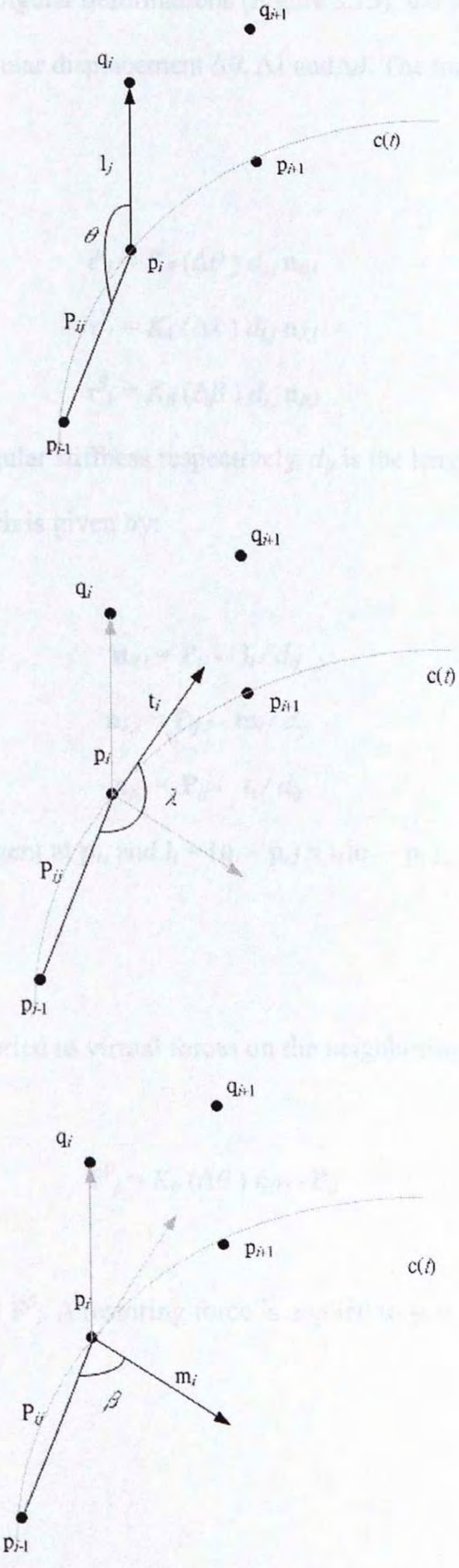


Figure 3.15 - The θ-, λ- and β- angular deformations (Top to bottom)

For the first three types of angular deformations (Figure 3.15), the torque at the point mass $\mathbf{p}_i$ is proportional to the angular displacement $\Delta\theta$, $\Delta\lambda$ and$\Delta\beta$. The torque can be expressed as:

$$\tau^{\theta}{}_{i} = K_{\theta}\,(\Delta\theta\,)\,d_{i,j}\,\mathbf{n}_{\theta,i}$$
$$\tau^{\lambda}{}_{i} = K_{\lambda}\,(\Delta\lambda\,)\,d_{i,j}\,\mathbf{n}_{\lambda,i}$$
$$\tau^{\beta}{}_{i} = K_{\beta}\,(\Delta\beta\,)\,d_{i,j}\,\mathbf{n}_{\beta,i} \qquad \text{.........}\ (3.12)$$

where $K_{\theta}, K_{\lambda}, K_{\beta}$ are the angular stiffness respectively, d_{ij} is the length of vector $\mathbf{P}_{ij}$ and $\mathbf{n}$ is the normal unit vector which is given by:

$$\mathbf{n}_{\theta,i} = \mathbf{P}_{ij} * \ \mathbf{l}_i / d_{ij}$$
$$\mathbf{n}_{\lambda,i} = \mathbf{P}_{ij} * \ \mathbf{m}_i / d_{ij}$$
$$\mathbf{n}_{\beta,i} = \mathbf{P}_{ij} * \ \mathbf{t}_i / d_{ij} \qquad \text{..........}\ (3.13)$$

where $\mathbf{t}_i$ is the polygon tangent at $\mathbf{p}_i$, and $\mathbf{l}_i = (\mathbf{q}_i - \mathbf{p}_i) \times \mathbf{t}_i\,|\mathbf{q}_i - \mathbf{p}_i\,|$,

$\mathbf{m}_i = \mathbf{t}_i \times \mathbf{l}_i$.

The torques are then converted to virtual forces on the neighboring point mass.

$$\mathbf{F}^{\theta}{}_{j} = K_{\theta}\,(\Delta\theta\,)\,\mathbf{n}_{\theta,i} * \mathbf{P}_{ij} \qquad \text{..........}\ (3.14)$$

The same holds for $\mathbf{F}^{\lambda}{}_{i}$ and $\mathbf{F}^{\beta}{}_{i}$. A restoring force is applied to $\mathbf{p}_j$ in order to balance the torque. That is $\mathbf{F}^{\theta}{}_{i} = -\mathbf{F}^{\theta}{}_{j}$.

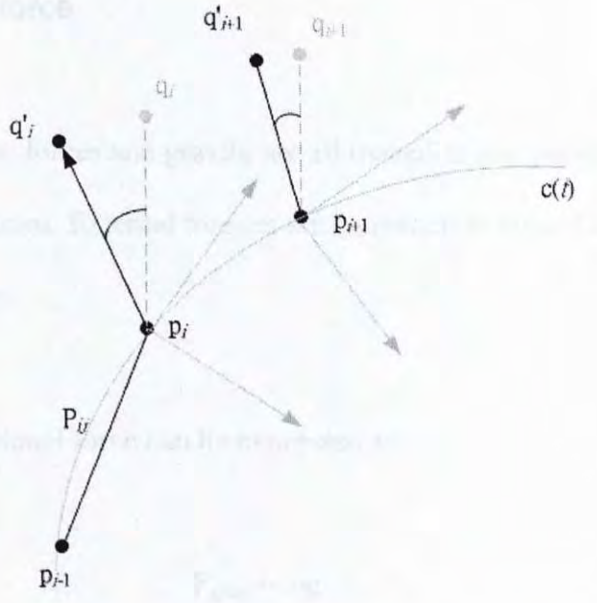


Figure 3.16 - The twisting deformation

For the twisting deformation on $\mathbf{P}_{ij}$, the torque at the point mass $\mathbf{p}_i$ is proportional to the angular displacement $\Delta\omega$ (Figure 3.16). The torque can be expressed as:

$$\tau^{\omega}{}_i = K_{\omega}\,(\Delta\omega)\,\mathbf{t}_i \qquad \text{.......... (3.15)}$$

where K_{ω} is the angular stiffness, d_{ij} is the length of vector $\mathbf{P}_{ij}$ and $\mathbf{t}_i$ is the tangent vector at $\mathbf{p}_i$ and $\Delta\omega = \omega_i - \omega_j$. The torque is then converted to virtual forces on the point mass $\mathbf{q}_i$ and $\mathbf{q}_j$ of the orientation curve and they are expressed as:

$$\mathbf{F}^{\omega}{}_i = K_{\omega}\,(\Delta\omega)\,\mathbf{t}_i * \mathbf{l}_i$$

$$\mathbf{F}^{\omega}{}_j = K_{\omega}\,(\Delta\omega)\,\mathbf{t}_j * \mathbf{l}_j \qquad \text{.......... (3.16)}$$

3.3.3 External force

User-interactions, reaction forces and gravity are all treated as independent external forces in the dynamics formulations. External torques are converted to virtual forces on the neighboring point masses.

For example, the gravitational force can be expressed as:

$$\mathbf{F}_{grav} = m\mathbf{g} \qquad \text{………. (3.17)}$$

where m is the mass of the object and $\mathbf{g}$ is the gravitational acceleration.

The force of air resistant can be expressed as:

$$\mathbf{F}_{air} = 0.5C_{drag}\mathbf{v}^2 \qquad \text{………. (3.18)}$$

where $\mathbf{v} = \mathbf{v}_{obj} - \mathbf{v}_{wind}$, and C is the dragging coefficient.

3.4 Equation of motion

Lagrangian formulation is used to model the dynamics of the mass-spring system. The Lagrangian equation of motion is expressed as:

$$m\mathbf{r}'' + d\mathbf{r}' + k\mathbf{r} = \mathbf{f}_{external} \qquad \text{.......... (3.19)}$$

for any point mass at location **r**. *m*, *d*, *k* are the mass, damping coefficient and the linear stiffness of the point mass respectively.

The first term represents the kinetic components of the total force while the second term represents the damping components. The third term *k***r** represents the internal force at the point mass. The internal forces at a point mass **r** which is $-[\,\mathbf{f}^{\delta}(\mathbf{r})+\mathbf{f}^{\theta}(\mathbf{r})+\mathbf{f}^{\lambda}(\mathbf{r})+\mathbf{f}^{\beta}(\mathbf{r})\,]$, which is the sum of external forces at the point mass at **r**.

The equation can be written in a matrix format with **R** represents a 2*N* x 3 position vector, that is, $\mathbf{R} = [\,\mathbf{p}_0 \quad \mathbf{p}_1 \ldots \mathbf{p}_{N-1} \quad \mathbf{q}_0 \quad \mathbf{q}_1 \ldots \mathbf{q}_{N-1}]^T$. *N* is half of the total number of point masses. As a result, we have:

$$M\mathbf{R}'' + D\mathbf{R}' + K\mathbf{R} = \mathbf{F}_{external} \qquad \text{.......... (3.20)}$$

where *M* and *D* are 2*N* x 2*N* diagonal mass and damping matrices respectively, *K* is the 2*N* x 2*N* symmetric sparse matrix and $\mathbf{F}_{external}$ is the 2*N* x 3 force vector state. With the damping item is ignored, the Eq. (3.11) can be simplified to:

$$M\mathbf{R}'' + K\mathbf{R} = \mathbf{F}_{external} \qquad \text{.......... (3.21)}$$

The tensile force on a point mass $\mathbf{p}_i$ can be generalized from Eq. (3.11) as:

$$\mathbf{F}^{\delta}(\mathbf{p}_i) = \Sigma_j^{\,N} K_{i,j} [\, (\, \mathbf{p}_i - \mathbf{p}_j \,) + \delta_{i,j}\, \mathbf{u}_{i,j} \,] \qquad \text{..........} \; (3.22)$$

where N is the number of neighboring point mass, $\mathbf{p}_j$ is the j-th neighboring point mass, $K_{i,j}$ is the spring constant, δ is the rest length of $\mathbf{P}_{ij}$ and $\mathbf{u}_{i,j}$ is the unit vector in the direction of $\mathbf{P}_{ij}$.

The torsional force on a point mass $\mathbf{p}_i$ can be generalized from Eq. (3.14) as:

$$\mathbf{F}^{\theta}(\mathbf{p}_i) = \Sigma_k^{\,N} K_{\theta} \,(\Delta\theta\,)\, \mathbf{n}_{\theta,k} * \mathbf{P}_{ki}\,, \qquad \text{for } i=0 \text{ or } i=M-1$$

$$\mathbf{F}^{\theta}(\mathbf{p}_i) = -\,\Sigma_j^{\,N} K_{\theta} \,(\Delta\theta\,)\, \mathbf{n}_{\theta,i} * \mathbf{P}_{ij} + \Sigma_k^{\,N'} K_{\theta} \,(\Delta\theta\,)\, \mathbf{n}_{\theta,k} * \mathbf{P}_{ki}\,, \quad \text{for } 0<i<M-1$$

$$\text{..........} \; (3.23)$$

where M is the total number of point mass $\mathbf{p}$, N and N' are the numbers of neighboring point masses, $\mathbf{p}_j$ and $\mathbf{p}_k$ are the j-th and $j+1$-th neighboring point masses.

The same holds for $\mathbf{F}^{\lambda}(\mathbf{p}_i)$ and $\mathbf{F}^{\beta}(\mathbf{p}_i)$.

The torsional force on a point mass $\mathbf{q}_i$, can be generalized from the equation (3.17) as:

$$\mathbf{F}^{\omega}(\mathbf{q}_i) = \Sigma_j^{\,N} K_{\omega}\, (\omega_i - \omega_j)\, \mathbf{t}_i * \mathbf{l}_i \qquad \text{..........} \; (3.24)$$

where N is the number of neighboring point mass, $\mathbf{q}_j$ is the j-th neighboring point mass.

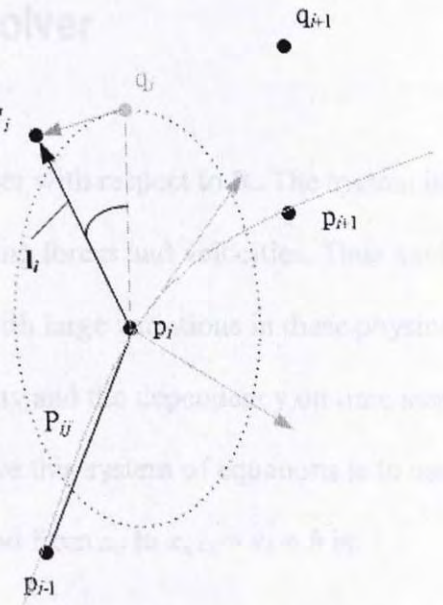


Figure 3.17 - Updating control points of the orientation curve

As the twisting deformation is governed by the orientation curve rotating along the primary curve (Figure 3.17), the $\mathbf{q}_i$' obtained from the equation is used to evaluate as ω_i' by following relationship:

$$\omega_i' = | \mathbf{q}_i' - \mathbf{q}_i | / | \mathbf{l}_i | \qquad \ldots\ldots\ldots (3.25)$$

Therefore the point mass $\mathbf{q}_i$ rotates for an angle $\omega_i' - \omega_i$.

3.5 System solver

The Eq. (3.21) is second order with respect to **R.** The system in our work handles a great number of equations involving forces and velocities. Thus a reliable scheme should be employed in order to deal with large variations in these physical terms. At the same time, the computational complexity and the dependency on time step should also be considered. The simplest method to solve this system of equations is to use the explicit Euler method. One step of the Euler method from x_n to $x_{n+1} = x_n + h$ is:

$$y_{n+1} = y_n + h f(x_n, y_n) \qquad \text{……….} (3.26)$$

The formula uses derivative information only at the beginning of the interval and all unknown values can be updated independently. However, the explicit Euler method leads to unlimited growth of the error in position and velocity of the mass and a step error $O(h^2)$ can be expected

On the other hand, the backward Euler method Eq. (3.27) is an implicit method which needs to solve an equation to solve y_{n+1}.

$$y_{n+1} = y_n + h f(x_{n+1}, y_{n+1}) \qquad \text{……….} (3.27)$$

The advantage of implicit methods such as Eq. (3.27) is that they are usually robust and more stable for solving stiff equation, meaning that a larger step size h can be used. Of course, it costs time to solve the equation and it is not always accurate as the result will contain errors such as Taylor-series truncation error.

Higher order implicit/explicit methods can further improve the stability as well as the accuracy. However, the computational complexity must be taken into consideration as this research simulates dynamic motion in real time. In computer animation, the accuracy is not so important when compares with engineering applications. As a result, a faster integrator should be considered.

Since high order implicit method will have much more calculations than the explicit one, an explicit Runge-Kutta method is adopted to solve the Eq. (3.21) in order to attain a satisfactory performance in stability, accuracy and speed. The computational accuracy and speed can be further improved if an adaptive step-size control scheme is applied.

The general form of a fourth-order Runge-Kutta formula is:

$$k_1 = Q(t_n, y_n)$$

$$k_2 = Q(t_n+h/2, y_n+h(k_1)/2)$$

$$k_3 = Q(t_n+h/2, y_n+h(k_2)/2)$$

$$k_4 = Q(t_n+h, y_n+h(k_3)/2)$$

$$y_{n+1} = y_n + h/6(k_1+2k_2+2k_3+k_4) + O(h^5) \qquad \text{.........} (3.28)$$

where h is the time step.

Eq. (3.21) can be expressed as a pair of first-order equations:

$$\mathbf{R}'' = (\mathbf{F}_{external} - K\mathbf{R}) / M$$

$$\mathbf{R}' = \mathbf{V}$$

$$\mathbf{V}' = \mathbf{F}_{total} / M \qquad \text{.........} (3.29)$$

where $\mathbf{F}_{total}$ is the sum of the internal and external forces.

These two equations are then solved by using the forth-order Runge-Kutta algorithm, that is, Eq. (3.28).

The Runge-Kutta method uses fixed step in the iteration. The results of prior solution are not used in its propagation. The solution may converge slowly and hence a large step size is preferred. On the other hand, a small step size is required to obtain a more accurate result when the system is changing rapidly. As a result, an adaptive algorithm is preferred.

The purpose of applying an adaptive step-size control is to achieve some predetermined accuracy in the solution with minimum computational effort. Implementation of adaptive step-size control requires an estimate of its performance and the truncation error based on information in the current iteration.

Fehlberg [20] proposed that given a fifth-order method with six function evaluations, another combination of the six functions gives a fourth-order method. The difference between the two estimates of $y(x+h)$ can be used as an estimate of the truncation error to adjust the step-size.

Given the general form of a fifth-order Runge-Kutta formula as

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n+a_2h, y_n+b_{21}k_1)$$

$$k_3 = hf(x_n+a_3h, y_n+b_{31}k_1+b_{32}k_2)$$

$$\dots$$

$$y_{n+1} = y_n + c_1k_1 + c_2k_2 + c_3k_3 + c_4k_4 + c_5k_5 + c_6k_6 + O(h^6) \qquad \text{..........} (3.30)$$

The embedded fourth-order formula is:

$$y^*_{n+1} = y_n + c^*_1k_1 + c^*_2k_2 + c^*_3k_3 + c^*_4k_4 + c^*_5k_5 + c^*_6k_6 + O(h^5) \quad \text{..........} \; (3.31)$$

Take the error estimate as

$$\varphi = y_n+1 - y^*_{n+1} = \Sigma^6_{i=1}\,(c_i - c_i^*)k_i \quad \text{..........} \; (3.32)$$

The coefficients proposed by Cash and Karp [21] are listed in Table 1.

i	a_i	b_{ij}					c_i	c_i^*
1							(37/378)	(2825/27648)
2	(1/5)	(1/5)					0	0
3	(3/10)	(3/40)	(9/40)				(250/621)	(18575/48384
4	(3/5)	(3/10)	(-9/10)	(6/5)			(125/594)	(13525/55296)
5	1	(-11/54)	(5/2)	(-70/27)	(35/27)		0	(277/14336)
6	(7/8)	(1631/55296)	(175/512)	(575/13824)	(44275/110592)	(253/4096)	(512/1771)	(1/4)
j		1	2	3	4	5		

Table 1 - Cash-Karp parameters for embedded Runge-Kutta method

If a step h_1 is taken to produce an error φ_1, then the step h_0 that would have given some other value φ_0 is estimated as

$$h_0 = h_1(\varphi_0 / \varphi_1)^{0.2} \quad \text{..........} \; (3.33)$$

If the magnitude of φ_1 is larger than φ_0, the equation indicates the amount to decrease the step-size in the next iteration.

However, if φ_0 has an implicit scaling with h, then the exponent is no longer correct. When the step-size is reduced from a large value, the new predicted value h_1 will fail to meet the desired accuracy. As a result, another exponent factor 0.25 has to be used. h_0 is re-defined as

$$h_0 = 0.93h_1(\varphi_0/\varphi_1)^{0.2} \qquad \text{if } \varphi_0>=\varphi_1$$

$$h_0 = 0.93h_1(\varphi_0/\varphi_1)^{0.25} \qquad \text{if } \varphi_0<\varphi_1 \qquad \text{………} \; (3.34)$$

3.6 Hierarchical representation

The axial skeletal representation described in the previous work conducted by Hui [9] allows using individual curve-pair for each desired part of the object and attaching the curve-pair to another one. The axial skeletal representation of an object is an inverted tree in which the motion and deformation of the parent nodes affects that of the child nodes.

To transform the object, a transformation is applied to the curve-pair of the root node. Manipulating the axial curve-pair of a component in the hierarchical structure of an object deforms the corresponding component while the other sub-components attached to the shape are not deformed. The sub-components may be transformed in response to the transformation of the parent curve-pair.

This provides an effective way for manipulating objects with individual deformable feature components. To extend the use of the deformation tool, we allow a curve-pair to affect its neighbors. As a result, a dynamic hierarchical axial curve-pair representation is presented in which a physics-based approach is adopted for manipulating objects. In this approach, the sub-component may affect the parent axial curve-pair of a shape component as discussed below.

In the section, we formulate the dynamic hierarchical representation of the dynamic axial curve-pair. The dynamics formulation gives rises to forces generating local deformations, and which may result in a global deformation.

The proposed approach is based on a mass-spring system. In order to attach a curve-pair to another one, the point mass $\mathbf{p_n}$ of a curve-pair is merged with the point mass $\mathbf{p_0}$' of another curve-pair to give a common point **p"**. A torsional spring is incorporated in connecting the points $\mathbf{q_n}$ and $\mathbf{p_1}$' as shown in Figure 3.18.

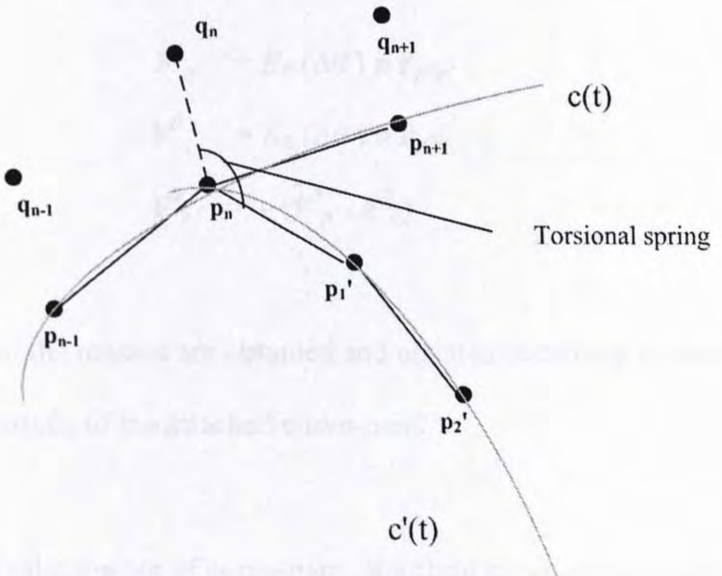


Figure 3.18 - Connection between two curve-pairs

Because of the hierarchical relationship of the curve-pairs as illustrated in Figure 3.18, the curve-pair **c** will be twisted when the curve pair **c'** is rotated about the tangent vector at $\mathbf{p_n}$. The torsional stress at $\mathbf{p_n}$ is proportional to the angular displacement $\Delta\theta$. The torque can be expressed as:

$$\tau^{\theta}_{p''} = K_{\theta}\,(\Delta\theta\,)\; d_{p''q}\; d_{p''p'}\,\mathbf{n} \qquad \text{..........} (3.35)$$

where K_θ is the angular stiffness, $d_{p''q}$ is the length of vector $\mathbf{P}_{p''q}$ and $d_{p''p'}$ is the length of the vector $\mathbf{P}_{p''p'}$.

n is the normal unit vector which is given by:

$$\mathbf{n} = \mathbf{P}_{p''q}\,\mathbf{P}_{p''p'} / (d_{p''q}\ \ d_{p''p'}) \qquad \text{.......... (3.36)}$$

The torque is then converted to virtual forces on the neighboring point masses.

$$\begin{aligned} \mathbf{F}^{\theta}_{p'} &= K_{\theta}\,(\Delta\theta)\,\mathbf{n}\,\mathbf{P}_{p''p'} \\ \mathbf{F}^{\theta}_{q} &= K_{\theta}\,(\Delta\theta)\,\mathbf{n}\,\mathbf{P}_{p''q} \\ \mathbf{F}^{\theta}_{p''} &= -\,(\mathbf{F}^{\theta}_{p'} + \mathbf{F}^{\theta}_{q}) \end{aligned} \qquad \text{.......... (3.37)}$$

Then the position of the masses are obtained and updated according to the tensile and twisting transformations of the attached curve-pairs.

In the parent-child relationship of curve-pairs, the child curve-pair is mapped to the axial space of its parent curve-pair. As a result, a child curve-pair will be transformed according to the local coordinate frame relative to the parent curve-pair. On the converse, a local transformation of the child node will not affect the parent.

In the hierarchical dynamic curve-pair representation, a primary curve-pair skeleton is considered to be the parent and the other attachments are identified as children. The parent and children would affect each other based on different configurations of the dynamics properties.

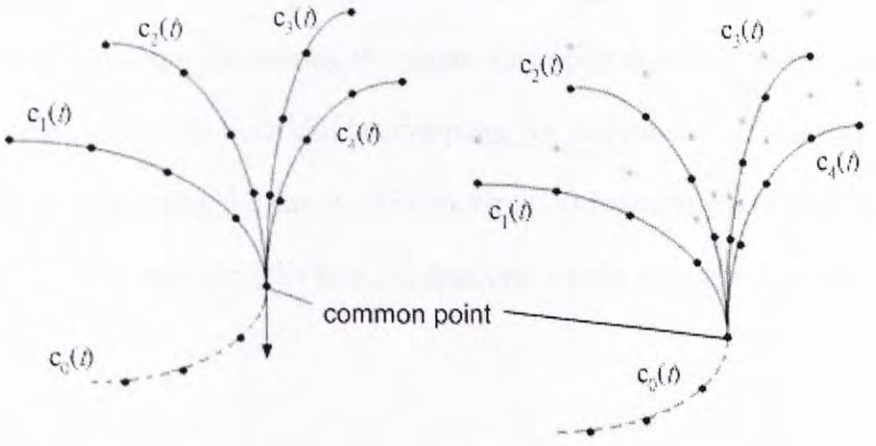


Figure 3.19 - The parent-child hierarchical representation: parent (dotted line), children (solid line)

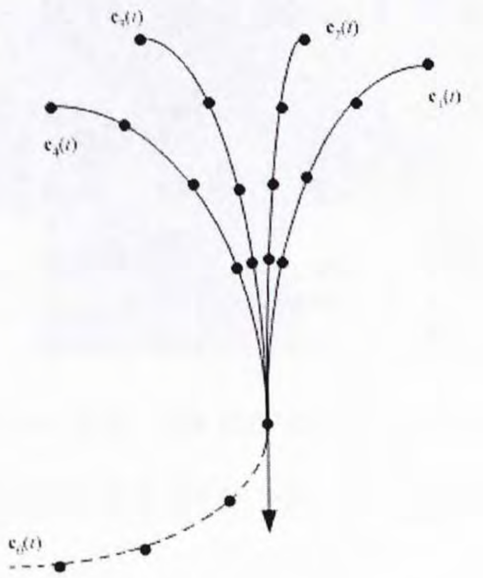


Figure 3.20 - The hierarchical dynamic curve-pair representation

The hierarchical dynamic curve-pair representation maintains the shape of the child curve-pair by adjusting corresponding spring configuration. In Figure 3.19, four child curve-pairs are attached to the parent curve-pair $\mathbf{c}_0(t)$. When the parent moves downwards, the children will move downwards with the shape of their skeleton remains unchanged.

In Figure 3.20, a trumpet shape is constructed easily using the hierarchical dynamic curve-pair representation. By moving the parent curve-pair downwards and minimizing the effect of angular springs on each child curve-pairs, the skeleton of the children are deformed as desired. Using the parent-child hierarchical representation, a similar result could be obtained by adjusting the four children curve-pairs instead of adjusting the parent only.

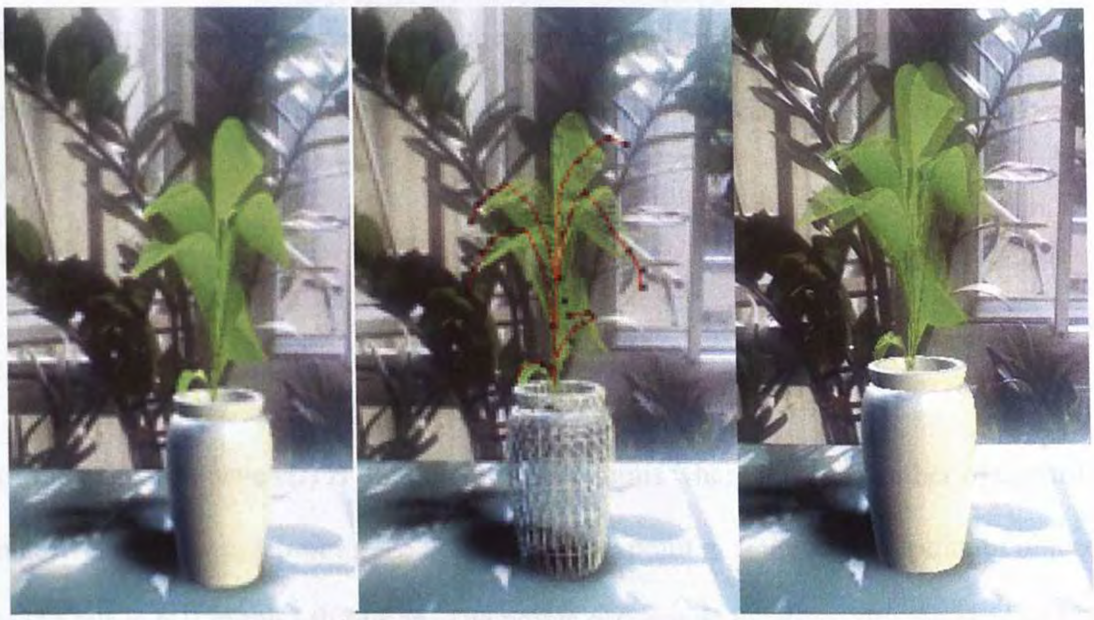


Figure 3.21 – The deformation of a plant: before deformation (left and middle), after deformation (right)

The technique is especially helpful in modeling plants. Figure 3.21 demonstrates the deformation of a plant with each leaf acts as a child to the parent, stem. When one children leaf is deformed by the wind, the stem will be bended accordingly as well as other leaves. The tool minimizes the design complexity when modeling different objects which are affecting each others.

3.7 Collision detection

A collision detection scheme for the axial curve-pair representation is introduced. In situation where collisions have to be detected between different meshes, bounding volume techniques built on the meshes are always an effective mean.

In this section, we presented the bounding cylinder approach. The bounding cylinder is built based on the mesh geometry and the attached axial curve-pair. When an object S is attached to an axial curve-pair, its vertex $\mathbf{s}_\mathrm{i}$ is defined in an axial space $A_c\{\mathbf{c}(t), \mathbf{c}_\mathrm{D}(t)\}$ as (t_i, u, v, w), that is:

$$\mathbf{s}_i = \mathbf{c}(t_i) + u\,\mathbf{l}_\mathrm{x}(t_i) + v\,\mathbf{l}_\mathrm{y}(t_i) + w\,\mathbf{l}_\mathrm{z}(t_i) \qquad \text{.......... (3.38)}$$

Next, the primary curve $\mathbf{c}(t)$ is divided into $n\text{-}1$ regions where n is the number of control points. In each region, a point $\mathbf{g} \in S$ is selected. The point $\mathbf{g}_i$ is the farthest point of object S from the curve $\mathbf{c}(t)$ in the i-th region. The points $\mathbf{g}_i$ $(i = n\text{-}1)$ together with the head and tail control points, $\mathbf{c}(0)$ and $\mathbf{c}(1)$, of the primary curve are used to construct a bounding line segments. Sweeping the line segment along the primary curve $\mathbf{c}(t)$ will create a cylinder-shape volume (Figure 3.22). This volume is called the bounding cylinder of a curve-pair. Any objects intersecting the volume are considered to be a colliding object.

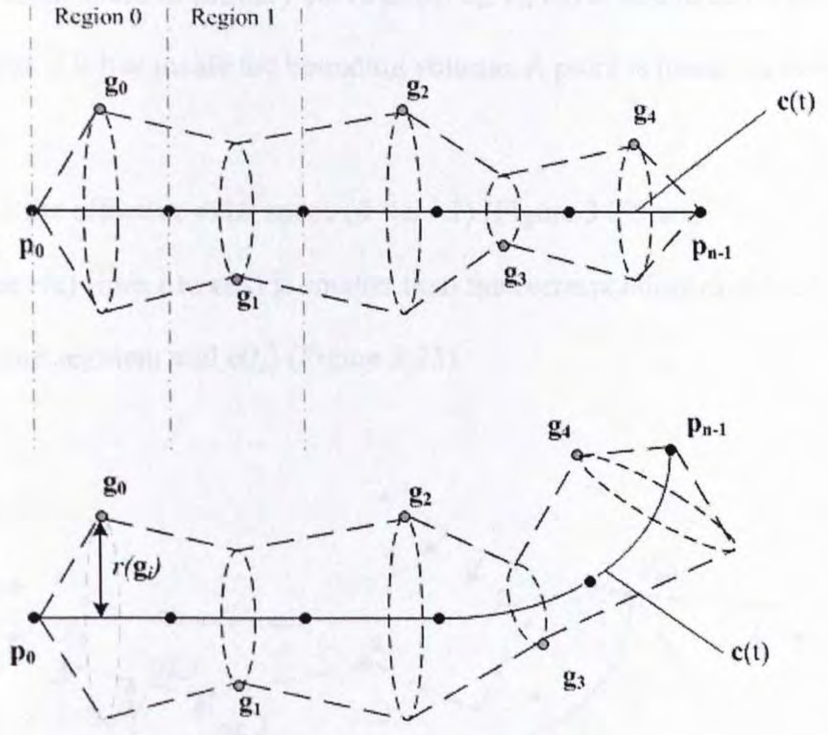


Figure 3.22 - The bounding cylinders

To determine the points $\mathbf{g}_i$ we first map all mesh points $\mathbf{s}_i$ in the axial space $A_c\{\mathbf{c}(t), \mathbf{c}_D(t)\}$ of the curve-pair. A point $\mathbf{s}_i$ is classified to a region of the primary curve according to its t value. The distance $r(\mathbf{s}_i)$ between $\mathbf{s}_i$ and the primary curve $\mathbf{c}(t)$ is equal to $(u^2 + v^2)^{1/2}$.

The bounding vertex $\mathbf{g}$ is selected from the points $\mathbf{s}_i$ in the i-th region and which is the farthest from the primary curve. Thus, we have $r(\mathbf{g}) = \max\{\ r(\mathbf{s}_i)\ \}$ where $\mathbf{s}_i$ are mesh points in the i-th region. In each region, we select one bounding vertex from $\mathbf{s}_i$ and if there is no mesh point in the region, a bounding vertex $\mathbf{g}$ will be created with the same $r(\mathbf{g})$ as the bounding vertex in previous region.

A point **e** in the axial space of primary curve as (t_e, u_e, v_e, w_e) is said to be collided with the bounding cylinder if it lies inside the bounding volume. A point is inside the bounding cylinder if

i) **e** lies inside the effective axial space $(0 < t_e < 1)$ (Figure 3.22) and

ii) the distance $r(\mathbf{e})$ from **e** to $\mathbf{c}(t_e)$ is smaller than the corresponding distance between the bounding line segment and $\mathbf{c}(t_e)$ (Figure 3.23).

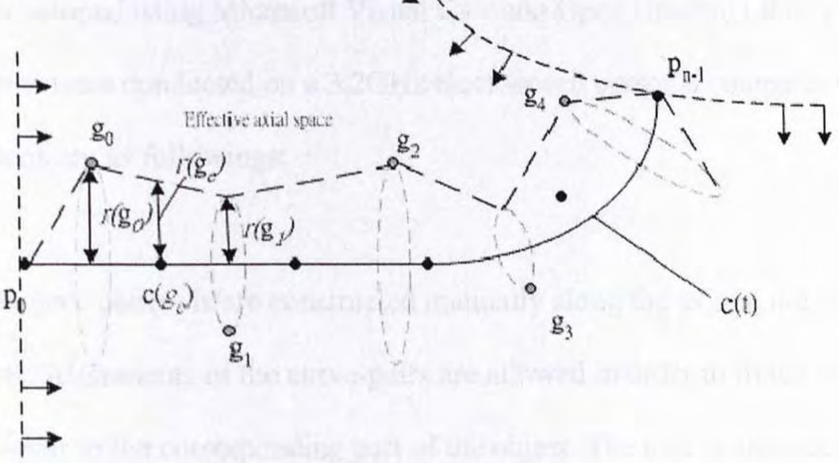


Figure 3.23 - The effective axial space

Given $\mathbf{g}_i$: $(t_i, u_i, v_i, w_i) \in A_c\{\mathbf{c}(t)\ \mathbf{c}_D(t)$, $\mathbf{g}_j$: $(t_j, u_j, v_j, w_j) \in A_c\{\mathbf{c}(t), \mathbf{c}_D(t)\}$ and a point $\mathbf{g}_e$ in the i-th region, the distance $r(\mathbf{g}_e)$ between the bounding line segment and $\mathbf{c}(t_e)$ is

$$r(\mathbf{g}_e) = (t_e - t_i / t_j - t_i)[r(\mathbf{g}_j) - r(\mathbf{g}_i)] + r(\mathbf{g}_i) \qquad \text{……….} (3.39)$$

where $\mathbf{c}(t_e)$ lies within $\mathbf{c}(t_i)$ and $\mathbf{c}(t_j)$ and $t_i < t_e < t_j$,

Chapter 4

Implementation and experimental results

An experimental system was implemented to simulate the deformation of an object. The system was developed using Microsoft Visual C++ and Open Graphic Library (OpenGL). The simulations were conducted on a 3.2GHz clock-speed personal computer with 2GB RAM. The steps are as followings:

i) Axial curve-pair(s) is/are constructed manually along the axis of the deformable objects. Alignments of the curve-pairs are allowed in order to fit the axial curve-pair to the corresponding part of the object. The user is allowed to control the spring constants, damping confidents and the mass properties of each element of the curve-pair model.

ii) Vertices on the deformable objects are then attached to the corresponding axial curve-pair that was aligned along the axial region of the objects manually.

iii) If a hierarchical structure of axial curve-pair is required, the relationship of the curve-pairs is defined.

iv) Constraints and external forces are added to the corresponding control points of the curve-pairs.

v) Once computed, the position information for each point masses of the driven mass spring system of each axial curve-pair at each simulation step is obtained. The axial curve-pairs are updated as well as the attached vertices.

4.1 Comparison with original mass-spring simulation

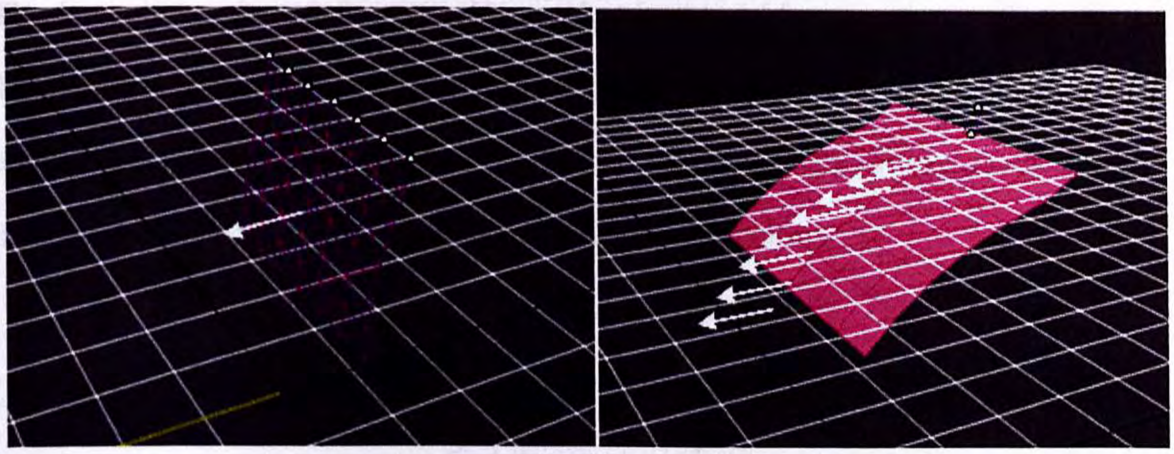


Figure 4.1 - A cloth simulation. Mass-spring system applied directly on the cloth (left). Mass-spring system adopted in the attaching axial curve pair (right)

With the same hardware platform and the same mass-spring system solver, we compared the results of the deformation simulation conducted in two different methods. On the left of Figure 4.1, the deformable object is a cloth which is modeled with a mass spring system. 144 springs are used to connect 49 vertices. On the right of Figure 4.1, a cloth model with the same vertices as the one used in the left, is attached to a dynamic axial curve-pair with 32 springs.

The same directional force is applied to both objects. The cloth deformation on the left gains a simulation speed of 34 frames per second while the speed is 76 frames per second for the cloth simulation on the right. The simulation speed is proportional to the number of masses and springs. Instead of applying dynamics directly to the mesh geometry, the dynamic axial curve-pair obtains a similar physical deformation with a faster speed.

4.2 Comparison with the dynamic free form deformation

The dynamic free form deformation [6] was first introduced for animation synthesis. The dynamic constraints are implemented at the control points. Through the deformation of control lattices, the embedded object is deformed correspondingly. As a result, the physical properties could not be fully reflected on the deformable objects.

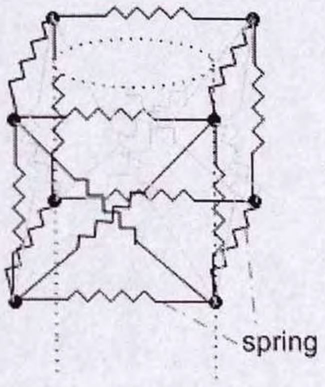


Figure 4.2 – A dynamic free form deformation lattice

The proposed dynamic axial curve-pair deformation tool deforms an embedded object by modifying the dynamic axial curve-pair. The dynamic axial curve-pair is fitted to a mass-spring system of which the physical properties can be expressed in the embedded object in a better way. Moreover, it is more intuitive to control the deformation by modifying the shape of the dynamic axial curve-pair rather than modifying a 3D volume in dynamic FFD, especially when implementing a twisting deformation.

4.3 Comparison with the axial curve-pair deformation

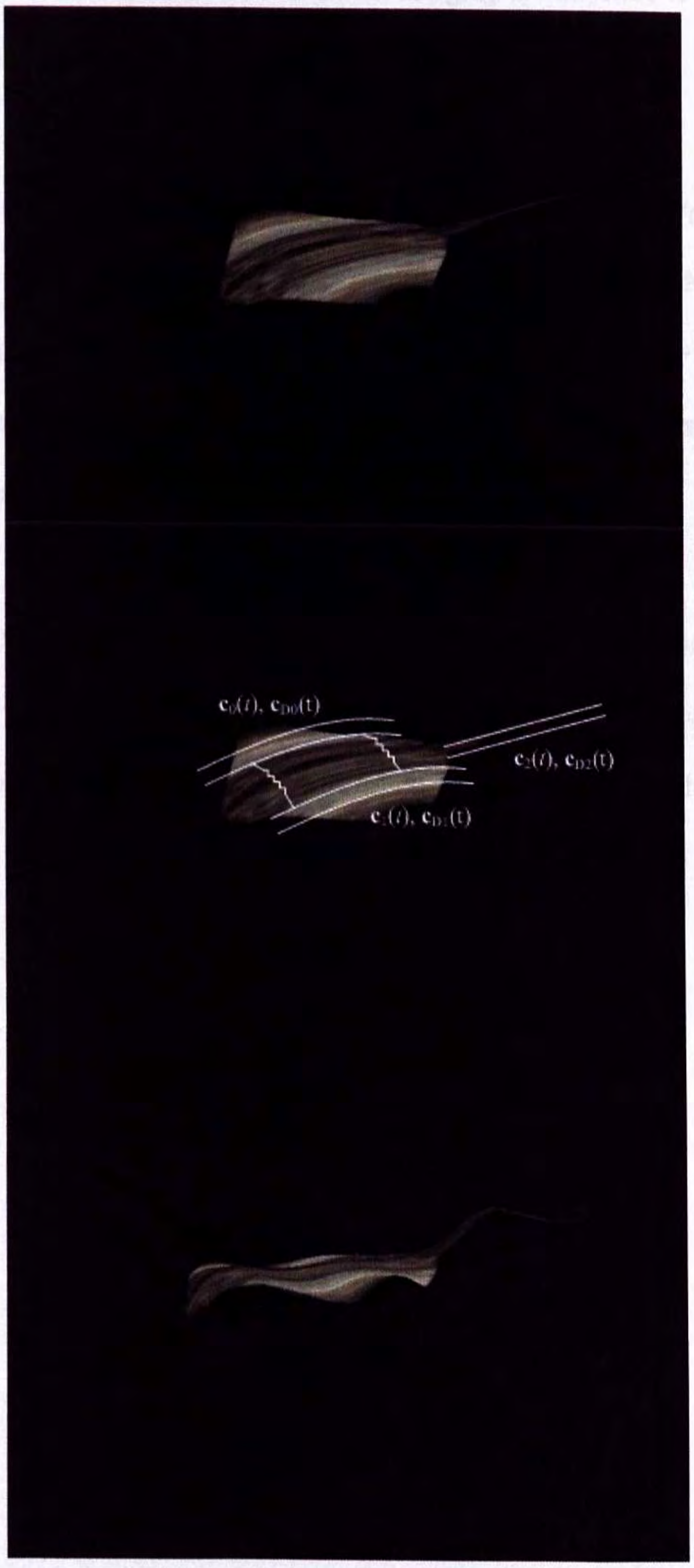


Figure 4.3 - A stingray swims with its fins and tail vibrating

In Figure 4.3, the swimming motion of a stingray is simulated. It illustrates how a hierarchical representation of dynamic axial curve-pair deformation influences the motion of a deformable object. The fins and the tail of the stingray are attached to their corresponding axial curve-pair.

The body of the stingray is controlled by a system of curve-pairs, $\{\mathbf{c}_0(t), \mathbf{c}_{D0}(t)\}$ and $\{\mathbf{c}_1(t), \mathbf{c}_{D1}(t)\}$. The two curve-pairs will affect each other in the bending deformation of the main body. The fins are vibrating by applying a sine function of force, $\mathbf{f} = A\mathbf{u}\, sin(t)$, on the point mass $\mathbf{q}_0$ of the orientation curves ($\mathbf{c}_{D0}(t)$ and $\mathbf{c}_{D1}(t)$). The tail vibrates due to a sine function of force $\mathbf{f}$ applied to the point mass $\mathbf{p}_0$ of the primary curve $\mathbf{c}_2(t)$.

Through defining external forces on the control points, an animation of the swimming motion can be easily obtained. Using the axial curve-pair deformation tool, every control point of the curve-pair has to be manually controlled in order to achieve the same result.

4.4 Shape restoring properties

The proposed method adopted a special designed mass-spring system which allows restoring the shape of embedded object if all dynamic constraints are removed. It is an ability which is useful in animating plants or cloth. However, the dynamic FFD which controls the twisting deformation by manipulating a group of control points does not guarantee a smooth restoring power in twisting deformation since the control points will restore to their initial setting independently instead of restoring in a group.

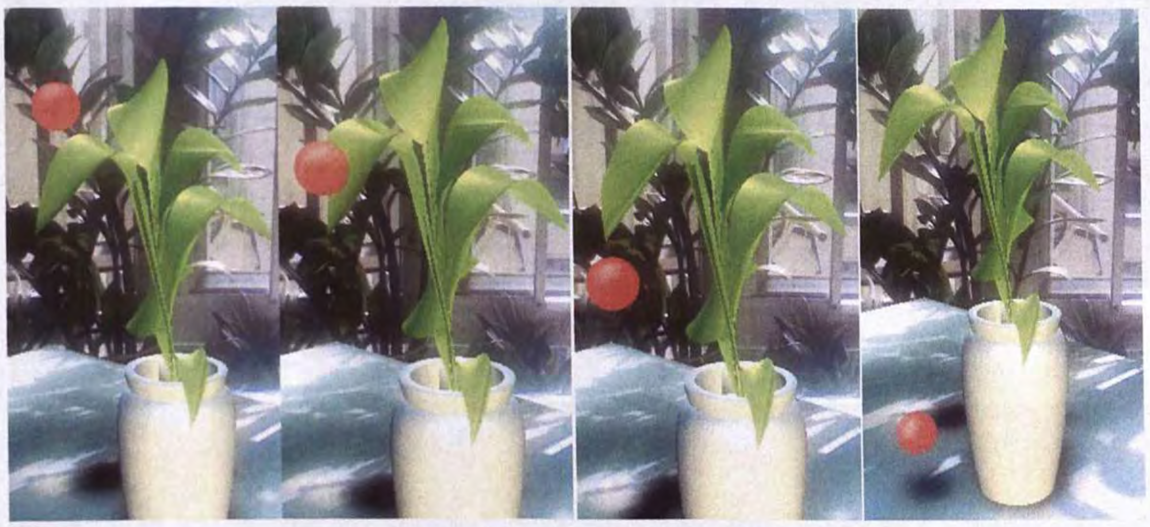


Figure 4.4 - The shape restoring ability of the dynamic axial curve-pair deformation: initial state (left), deformed state (middle-left and middle-right), restored state (right)

In Figure 4.4, a plant is associated with the dynamic axial curve-pairs. The initial shape of the plant is shown in the left picture with the leaves are kept crooked. The curve pairs are manually aligned to fit the axial of each deformation element, the leaves and the stem. A ball is dropped from a certain height and falls on the leaf. The leaf is then deformed when collides with the ball. It restores to its initial shape after being pressed down by the ball.

4.5 Applications

The proposed deformation method is useful in animating plants and cloths. These objects are mainly animated due to the dynamic constraints in the environment. For a plant, it usually consists of a number of leaves, a stem and root. The hierarchical representation of dynamic axial curve-pair is particularly useful in simulating plants.

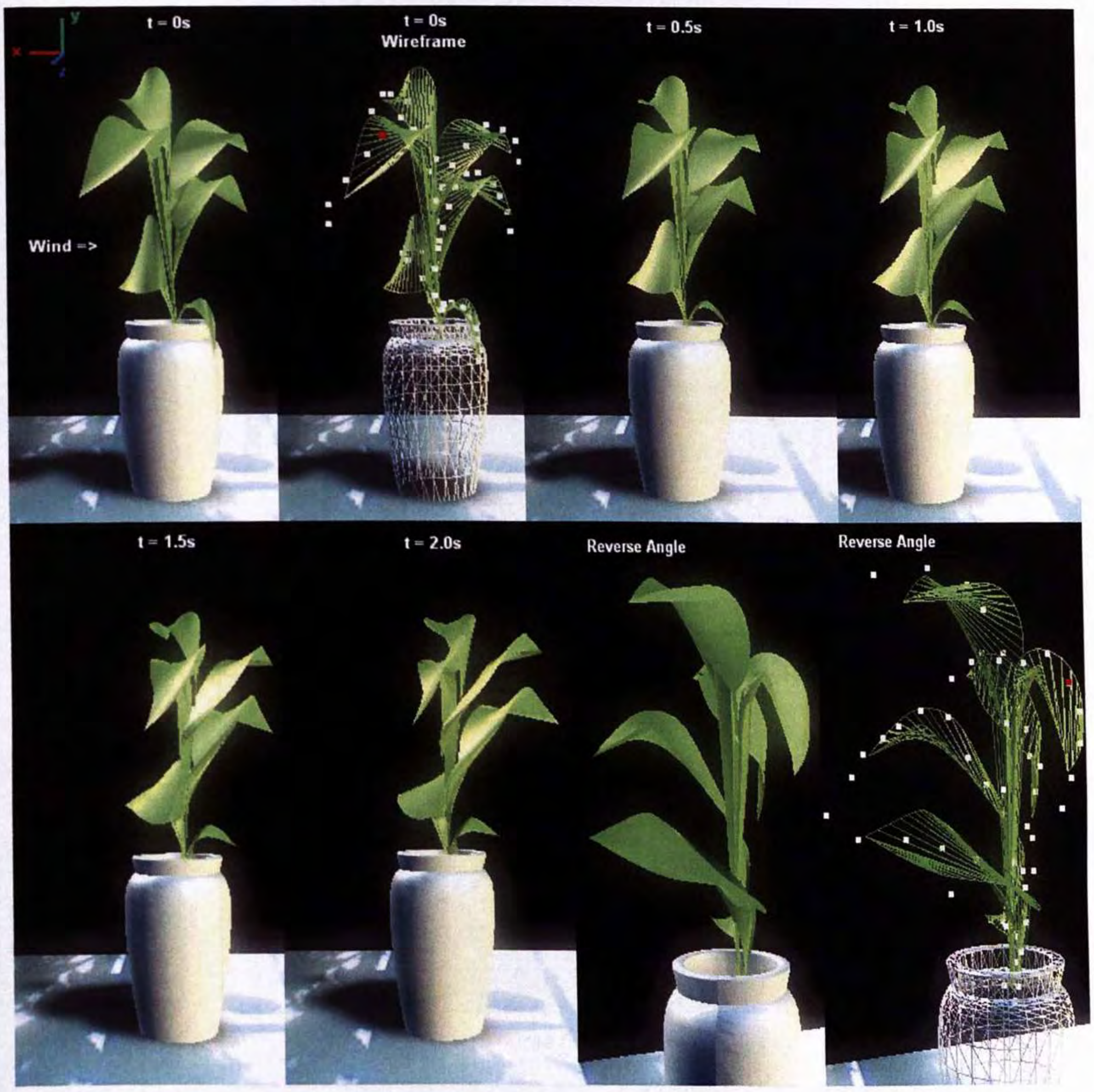
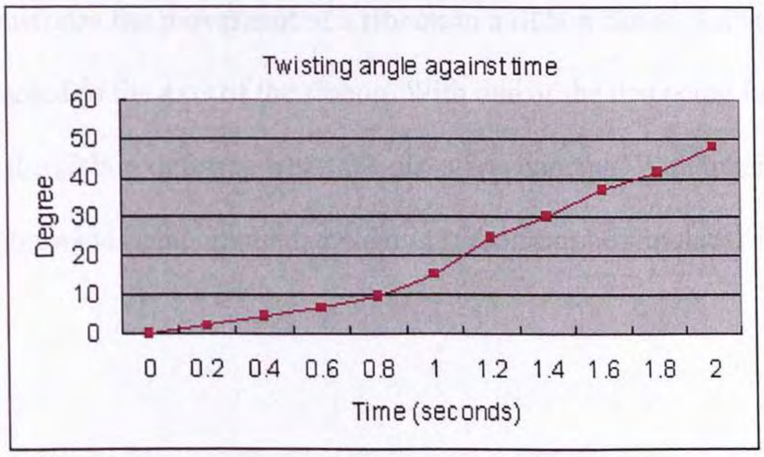


Figure 4.5 – Deformation of a plant under windy environment using dynamic hierarchical representation of dynamic axial curve-pairs

Figure 4.5 shows the leaves swaying in the wind gently. Snapshots are taken every 0.5 second and the last two shots view the plant in a reverse angle. A parent axial curve-pair is attached to the stem and the children axial curve-pairs are assigned to the leaves. The leaves bend and twist according to different dynamic constraints such as directional windy forces. As the leaves are being pulled or pushed, the stem and other siblings will be deformed correspondingly.

Refer to 4.5, the motion of the control point which is high-lighted in red is recorded and plotted against time as followings:



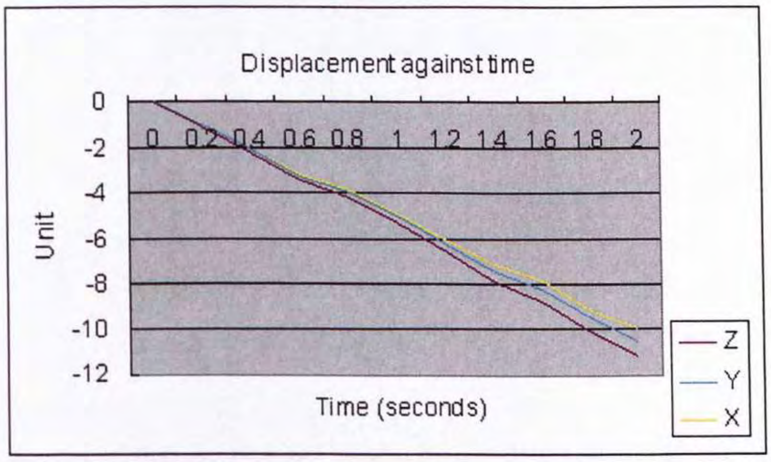


Figure 4.6 Motion of a control point (red-colored, Figure 4.5) during deformation: Twisting angle against time (upper) and Displacement against time (bottom)

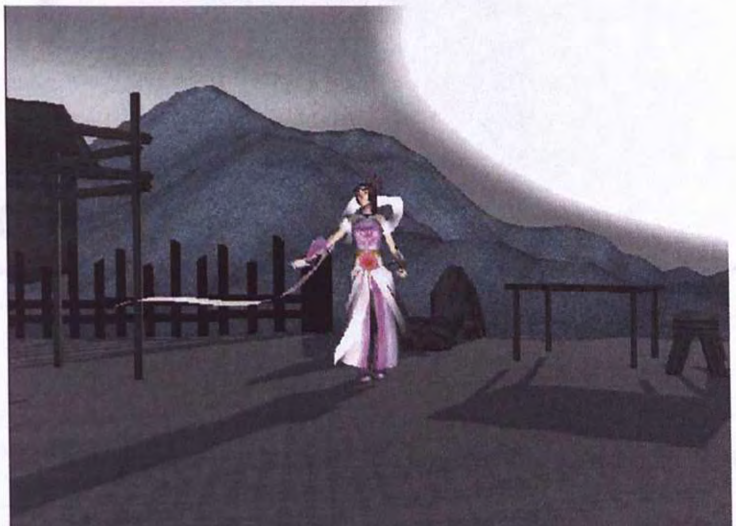


Figure 4.7 - Simulating ribbon movement in a ribbon dance

Figure 4.6 demonstrates the movement of a ribbon in a ribbon dance. A dynamic axial curve-pair is attached to the axis of the ribbon. With one of the tips being fixed to the right hand of the girl, the ribbon deforms when the girl starts dancing. With other constraints such as gravity and wind being applied, a waving ribbon can be simulated.

Chapter 5

Conclusion

We proposed and implemented a framework for physics-based deformation. Our approach adopts the axial curve-pair deformation and extends the technique for dynamic simulation. The intuitive stretching, bending and twisting abilities of axial curve-pair deformation technique provide a very good framework for free-form design. The technique provides a solution to the traditional axial deformation technique where lack of control on the local coordinate frame may lead to unexpected twist of the object.

With a well defined local coordinate frame, the axial curve-pair deformation tool can achieve stretching, bending and twisting deformations in a much easier and predictable way. Although the technique allows complex deformations to be performed, creating an animation using the axial curve-pair based deformation tool may be tedious because a large number of control points are required to be manipulated. Therefore, the system is further enhanced by incorporating a dynamic setting in this research.

A mass-spring model is adopted to model a physics-based axial curve-pair skeleton. Fitting a curve-pair to the point masses of the mass spring system emulates the properties of a physical deformation, and at the same time, a smooth deformation can be maintained.

By using special torsional spring elements to connect the local coordinate frames, mass-spring system is built on the curve-pair with smooth bending and twisting behaviors. Physics-based deformation is performed on the point masses, and hence deforming the embedded curve-pair.

Our technique is useful for designing physical meaningful shapes, and is useful for producing a physics-based deformation of animals and characters. This approach does not require users to manipulate large number of degrees of freedom, or control points, of the axial curve-pair. Instead, design operation can be performed in terms of forces and geometric constraints.

There are a number of enhancement of this research can be done the future. A better point projection method can be considered when attaching the model to the axial curve-pair. A better mapping scheme is required if a vertex to be attached results in two or more closest projections on the primary curve. The bounding cylinder technique proposed in the work is only effective when the embedded object is a cylinder-shape one. Flat objects may lead to incorrect collision prediction as the bounding cylinder may not give a correct estimate of the geometric information of the embedded object. The location and orientation of the curve-pair in our work is defined manually. Designers are required to initialize the geometric information of the curve-pair at the very beginning. An effective axis-searching algorithm will further enhance the system.

Reference

[1] Sederberg TW, Parry SR,, "Free-form deformation of solid geometric models", ACM Computer Graphics (SIGGRAPH '86), 1986.

[2] Borrel P., "Simple constrained deformations for geometric modeling and interactive design", ACM Transactions on Computer Graphics, vol.13, no.2, pp. 137-55, 1994.

[3] D. Terzoppoulos and H. Qin, "Dynamic NURBS with geometric constraints for interactive sculpting", ACM Transactions on Graphics, vol.13, no.2, pp. 103-136, April 1994.

[4] J. E. Chadwick, D. R. Haumann, and R. E. Parent, "Layered construction of deformable animated characters", Proceedings of ACM SIGGRAPH, Computer Graphics, vol. 23, no. 3, pp. 243-252, July 1989.

[5] J. Christensen, J. Marks, and J. T. Ngo, "Automatic motion synthesis for 3D mass-spring models", Tech.Rep., MERL TR95-01, 1995.

[6] P. Faloutsos, M. van de Panne, D. Terzopoulos, "Dynamic Animation Synthesis with Free-Form Deformations", IEEE Transactions on Visualization and Computer Graphics, 1997.

[7] J.Q.Feng, L.Z.Ma and Q.S.Peng, "A New Free-form Deformation through the Control of Parametric Surfaces", Computers&Graphics, 20(4), 531-539 (1996)

[8] Lazarus F., Conquillart S., Jancene P., "Axial deformations: an intuitive deformation technique", Computer-Aided Design, 26(8), pp. 603-617, 1994.

[9] K.C. Hui, "Free-form design using axial curve-pairs", Computer-Aided Design 34 (2002) 583-595.

[10] Hong Qin, D. Terzopoulos, "D-NURBS: A physics-based framework for geometric design", IEEE Transactions on Visualization and Computer Graphics, vol 2, no.1, March 1996

[11] P. Volino, N. Magnenat-Thalmann, "Comparing Efficiency of Integration Methods for Cloth Animation", Proceedings of Computer Graphics International '01, Hong-Kong, China, pp. 265–274, 2001.

[12] K.-J. Choi and H.-S. Ko, "Stable but responsive cloth", Proceedings SIGGRAPH '02, San Antonio, TX, USA, pp. 604–611, 2002.

[13] D. Baraff, A. Witkin, M. Kass, "Untangling Cloth", Proceedings SIGGRAPH '03, San Diego, CA, USA, pp. 862–870, 2003.

[14] B. Eberhardt, A. Weber, W. Strasser, "A Fast Flexible Particle-System Model for Cloth Draping", IEEE Computer Graphics and Applications, Vol. 16, No. 5, pp. 52–59, Sept.1996.

[15] D.R. Haumann, J. Wejchert, K. Arya, and B.Bacon, "An application of motion design and control in physically-based animation," Proceeding of Graphics Interface '91, pp. 279-286, 1991

[16] Q.H. Peng, X.G. Jin and J.Q. Feng, "Arc-Length-Based Axial Deformation and Length Preserving Deformation" Computer Animation'97, 1997, Geneva, IEEE Computer Society, 86-92.

[17] Faux ID, Pratt MJ., "Computational geometry for design and manufacturing", Wiley, 1979.

[18] Klok F., "Two moving coordinate frames for sweeping along a 3D trajectory", Computer-Aided Geometric Design, vol 3, pp 217-229, 1986.

[19] Lossing DL, Eshleman AL, "Planning a common data base for engineering and manufacturing", SHARE XLIII, Chicago, August 1974.

[20] Erwin Fehlberg, "Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems", NASA Technical Report 315, 1969

[21] Cash JR, Karp AH, "A variable order Runge–Kutta method for initial value problems with rapidly varying right-hand sides", ACM Trans Math Software 1990;16:201–222.

[22] G. Farin, "Curves and surfaces for computer aided geometric design: a practical guide", second edition, Academic Press, 1990.

[23] B.R. Gossick, "Hamilton's principal and physical system", New York and London: Academic Press, 1967.

[24] Hong Qin, D. Terzopoulos, "Triangular NURBS and their dynamic generalizations", Computer Aided Geometric Design, 1996

[25] D Metaxas, D. Terzopoulos, "Dynamic Deformation of Solid Primitives with Constraints", Computer Graphics, vol. 26, no. 2, pp. 309-312, 1992.

[26] Karan Singh, Eugene Fiume, "Wires: A Geometric Deformation Technique", Proceedings of SIGGRAPH'98, ACM Press, New York, p.405-414.

[27] P. Schneider, "Solving the Nearest-Point-on-Curve Problem", Graphics Gems, Academic Press, vol.1:607-612, 1990.

[28] Hsu,W., Hughes,J., Kaufmann,H., "Direct manipulation of free-form deformations", Computer Graphics, ,1992, 26:177-184.

[29] D. Bechmann, "Space Deformation Survey", Computer & Graphics, Vol. 18, No.4, pp571-586, 1994

[30] P. Borrel and D. Bechmann, "Deformation of N-dimensional objects", Int. J. Computational Geometry Appl., Vol. 1, No. 4, pp427-453, 1991

[31] A.H. Barr, "Global and local deformation of solid primitives", Siggraph '84, ACM Computer Graphics, vol. 18, No.3, pp21-34, 1984

[32] Y. Kho and M. Garland, "Sketching Mesh Deformation", Proceedings of the 2005 symposium on Interactive 3D graphics and games, pp147 – 154, 2005

[33] J. Lawrence and T.A. Funkhouser, "A painting interface for interactive surface deformation", Pacific conference on computer graphics and applications, 2003

CUHK Libraries
004546727