A Technique for Constructing Developable Surfaces

by

Meng Sun

A thesis submitted in conformity with the requirements
for the Degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Because real paper is approximately unstretchable, the surfaces obtained by bending a sheet of paper can be rolled out onto a plane without stretching or tearing. More precisely, such a rolling out maps the surface onto the plane, after which the length of any curve drawn on the surface remains the same. Such surfaces, when sufficiently regular, are well known to mathematicians as *developable surfaces*.

Developable surface modelling is mainly used for engineering purposes in Computer Aided Design (CAD). In this thesis, our focus is on using developable surface to model artistic objects which we see in everyday life.

We formulate a new developable surface representation technique. We describe a way of approximating a developable surface using a piecewise continuous tensor-product Bézier surface, provided that the developable surface has certain special properties. The feasibility of our model is demonstrated by applying it to the modelling of a hanging scarf and a bow. Possible extensions and interesting areas of further research are discussed.

Acknowledgements

我想借這個机會感謝我的導師 (Prof. Eugene Fiume) 和我的第二讀者 (Prof. Michiel van de Panne)。 感謝他們熱心的幫助和指導。

希望爸爸媽媽喜歡我用電腦勾劃的幼稚的圖像。

在傾心研究 "defeatable surfaces" 的几個月中，閱覽過的餐館菜單比科研文章還多，但在談笑中，時間的變量推導出的是華麗的數學公式表達不清的友情。

愿這短暫的瞬間，在師長與友人記憶的扉頁上留下一個快意的微笑。

Contents

List of Figures

Chapter 1

Introduction

Smooth surfaces must be generated in many computer graphics applications. Many real-world objects are inherently smooth, and much of computer graphics involves modelling the real world. The need to represent surfaces arises in two cases: in modelling existing objects (cars, faces, mountains) and in modelling virtual objects, where no preexisting physical object is being represented.

In the first case, a mathematical description of the object may be unavailable. Mathematically, one could use the infinitely-large set of points making up the object as a model, but this is not feasible for a computer with finite storage. More often, we merely approximate the object with pieces of planes, spheres, or other shapes that are easy to describe mathematically, and we require that points on our model be close to corresponding points on the object.

In the second case, when there is no preexisting object to model, the user creates the object in the modelling process; hence the object matches its representation exactly, because it is defined through the representation. To create the object, the user may sculpt the object interactively, describe it mathematically, or give an approximate description to be "filled in" by some program. In CAD, the computer representation is used later to generate physical realizations of the designed object.

A largely unexplored area in Computer Graphics consists of modelling area-constrained surfaces. For instance, when a flag is moving in the wind, the surface area of the flag should remain constant. To the author's knowledge, no one has constructed a modelling system which directly addresses such constraints. Previous work exists for curves which satisfy arc-length constraints [5] and objects which satisfy volume constraints, but there has been relatively little work on modelling surfaces which satisfy area constraints.

This thesis surveys the different approaches developed to design and model *developable surfaces*, which is an important subclass of area-constrained surfaces.

1.1 Why Developable Surfaces?

In order to construct a shape using a computer, it is necessary to produce a computer-compatible description of that shape. The exploration of the use of parametric curves and surfaces can be viewed as the origin of Computer Aided Geometric Design (CAGD). The major breakthroughs in CAGD are the theory of Bézier surfaces and Coons patches, later combined with B–spline methods.

Because real paper is approximately unstretchable, the surfaces obtained by bending a sheet of paper can be rolled out onto a plane without stretching or tearing. More precisely, such a rolling out maps the surface onto the plane, after which the length of any curve drawn on the surface remains the same. The surface is thus called isometric to the plane — “applicable” for short. Such surfaces, when sufficiently regular, are well known to mathematicians as *developable surfaces.*

As will be seen in chapter 2, a developable surface is the envelope of a one-parameter family of planes. While planar surfaces and completely general surfaces have a zero-parameter and a two-parameter family of tangent planes, developable surfaces have a one-parameter family of tangent planes [11]. A developable surface thus offers a complexity that is midway between that of a completely general surface and that of a plane surface. Modelling developable surfaces is important because it is a logical precursor to modelling area-constrained general surfaces.

1.2 Thesis Overview

This thesis begins by introducing the basic concepts and properties of developable surfaces. Since Bézier curves and surfaces are used in our modelling tool, related definitions and concepts are presented in chapter 3. In the past, three different types of approaches have been proposed to design and model developable surfaces. The mathematical reasoning and the algorithms will be discussed in detail in chapter 4. We will discuss the relationship between these existing approaches and our new technique. We subsequently present our new technique. Chapter 5 is devoted to explaining the new approach. An implementation of the new modelling tool is presented. Modelling issues such as local and global control are discussed. In chapter 6, examples of developable surfaces produced by this tool are used to demonstrate the capabilities and advantages of the new modelling system. The concluding chapter evaluates the new approach and the compromises it makes.

1.3 About the New Technique

In this thesis, we formulate a new technique of designing and modelling developable surfaces. Our goal is to give users good local and global control over the shape of the resulting surfaces. Our contribution is a design and implementation of such a system.

The new technique takes advantage of the geometric characteristics of the surface to be modelled. First, the surface to be modelled is divided into several pieces. Then the shape of each surface piece is defined by a generalized cone, where the shape of the cone is defined by its cross section together with its apex. The cross section of the cone can be designed interactively using piecewise continuous curves. Each surface piece is constructed separately in 3D. Then surface pieces are stitched together to construct the desired developable surface.

To obtain a better approximation of the desired developable surface, it is not necessary to subdivide each surface piece into smaller and smaller surface patches. To change the shape of one surface piece, the user can adjust control points of the cross section curve of the cone associated to this surface piece. Also, by editing a few controls points, the user can also alter the orientation of the tangent plane at a shared boundary of two adjacent surface pieces. Therefore, this approach gives the user a great deal of control over the shape of the resulting developable surface, both locally and globally. We will discuss this in detail in chapter 5.

Chapter 2

Developable Surfaces

2.1 Definitions

The definitions of ruled surfaces and developable surfaces are given in [2] as follows. A (differentiable) one-parameter family of (straight) lines $\{\alpha(t), \beta(t)\}$ is a correspondence that assigns to each $t \in I$ a point $\alpha(t) \in \mathbb{R}^3$ and a vector $\beta(t) \in \mathbb{R}^3$, $\beta(t) \neq 0$, where both $\alpha(t)$ and $\beta(t)$ are differentiable with respect to t, where I is the domain, $[a, b]$, of t, and $a, b \in \mathbb{R}$. For each $t \in I$, the line L_t which passes through $\alpha(t)$ and is parallel to $\beta(t)$ is called *the line of the family at* t. Given a one-parameter family of lines $\{\alpha(t), \beta(t)\}$, the surface

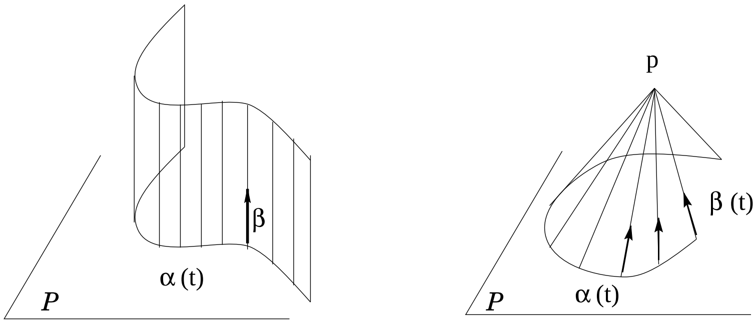
$$\mathbf{X}(t, v) = \alpha(t) + v\beta(t), t \in I, v \in \mathbb{R}, \tag{2.1}$$

is called the *ruled surface* generated by the family $\{\alpha(t), \beta(t)\}$. The lines L_t are called the *rulings*, and the curve α is called a *directrix* of the surface $\mathbf{X}$.

A ruled surface is said to be *developable* if

$$< \beta \times \frac{d\beta}{dt}, \frac{d\alpha}{dt} > \equiv 0, \tag{2.2}$$

i.e., β, $d\beta/dt$ and $d\alpha/dt$ are coplanar, for all points on the surface, where $< \beta \times \frac{d\beta}{dt}, \frac{d\alpha}{dt} >$ denotes the inner product of vector $\beta \times \frac{d\beta}{dt}$ and $\frac{d\alpha}{dt} \in \mathbb{R}^3$. The simplest



(a) a generalized cylinder (b) a generalized cone

Figure 2.1: Simple examples of developable surfaces.

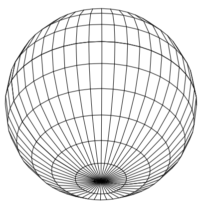


Figure 2.2: An example of a non-developable surface.

examples of developable surfaces are the cylinders and the cones. A generalized cylinder is a ruled surface generated by a one-parameter family of lines $\{\alpha(t), \beta(t)\}, t \in I$, where $\alpha(I)$ is contained in a plane P and $\beta(t)$ is parallel to a fixed direction in $\mathbb{R}^3$ (Figure 2.1(a)). For generalized cylinders, $dB(t)/dt = 0$. Note that the curve $\alpha(t)$ does not have to be a closed curve. A generalized cone is a ruled surface generated by a family $\{\alpha(t), \beta(t)\}$, $t \in I$, where $\alpha(I) \subset P$ and the ruling L_t all pass through a point $p \notin P$ (Figure 2.1(b)). Note that $d\alpha/dt = 0$ is a sufficient condition for such a family of curves to generate a generalized, but it is not a necessary condition. An example of a non-developable surface is a sphere (Figure 2.2). A wrinkled T-shirt is an example of an area-constrained surface which is not developable.

2.2 Properties of Developable Surfaces

In chapter 5, we will introduce our developable surface representation technique. This new technique takes advantage of properties of developable surfaces. Before describing these properties, we will introduce a number of terminologies which will be used later in this section: (1) the *intrinsic equation* of a curve in 2D, (2) the *osculating plane*, *normal plane* and *rectifying plane* of a space curve in 3D, (3) the *spherical image*. We will use these concepts to give an intuitive explanation of developable surfaces.

2.2.1 Intrinsic Equations of a Curve in $\mathbb{R}^2$

Once an initial point of a curve has been defined, the variation with the arc length s of the angle ψ subtended by its tangent on the x-axis is sufficient to define the curve in 2D, as shown in Figure 2.3. A relation between s and ψ is called an *intrinsic*

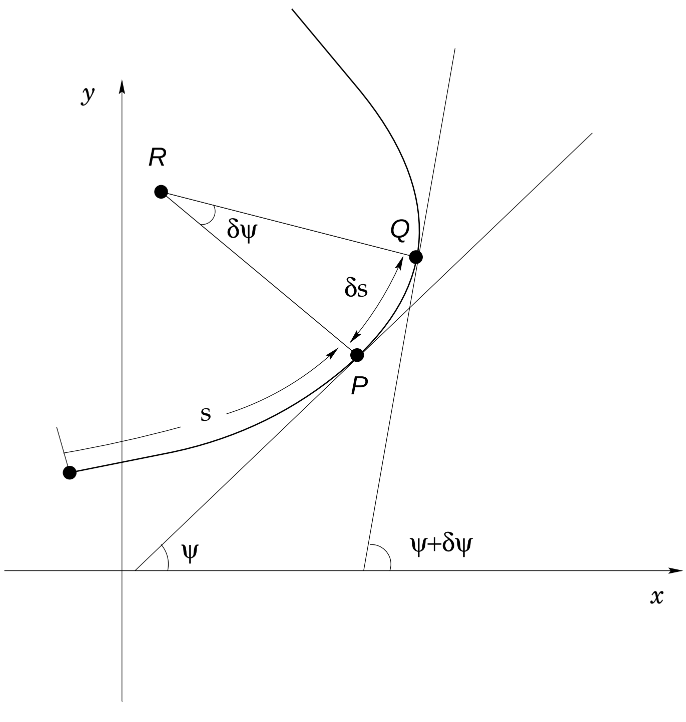


Figure 2.3: Define a 2D curve using s and ψ.

equation of the curve. Many intrinsic equations have the form $s = s(\psi)$.

The *curvature* κ is obtained from the intrinsic equation by the formula

$$\kappa = \frac{d\psi}{ds}. \tag{2.3}$$

Alternatively, the curve may be described parametrically in terms of the arc length by the equations $x = x(s)$ and $y = y(s)$. The functions $x(s)$ and $y(s)$ are then related to ψ by the equations

$$\frac{dx}{ds} = \cos\psi \tag{2.4}$$

and

$$\frac{dy}{ds} = \sin\psi. \tag{2.5}$$

If we differentiate these equations with respect to s, and substitute κ for $\frac{d\psi}{ds}$, $\frac{dx}{ds}$ for $\cos\psi$ and $\frac{dy}{ds}$ for $\sin\psi$, we obtain the simultaneous differential equations

$$\frac{d^2x}{ds^2} + \kappa(s)\frac{dy}{ds} = 0$$

$$\frac{d^2y}{ds^2} - \kappa(s)\frac{dx}{ds} = 0. \tag{2.6}$$

These two second order equations can in principle be solved to determine $x(s)$ and $y(s)$ for any given curvature function $\kappa(s)$. Effective numerical procedures have been described by Nutbourne (1972) [10] and Adams (1975).

2.2.2 Osculating Plane, Normal Plane and Rectifying Plane

Nonplanar curves in space are often referred to as *twisted curves*. Consider a small piece c of a general space curve over which the curve does not intersect itself and at each point of which the curve is smooth and well behaved, as shown in Figure 2.4. As in the case of a plane curve in $\mathbb{R}^3$, the *tangent* to c at point P on c is defined as the limiting position of the secant PQ as Q approaches P along the curve. Let $\mathbf{T}$ denote the unit tangent vector of c at P. Assume the twisted curve is parametrized in terms of t, $t \in \mathbb{R}$. The unit vector $\mathbf{N}$ in the direction of $\dot{\mathbf{T}} = d\mathbf{T}/dt$ is known as the *principal normal* vector. The vector product $\mathbf{T} \times \mathbf{N}$ defines a third unit vector perpendicular

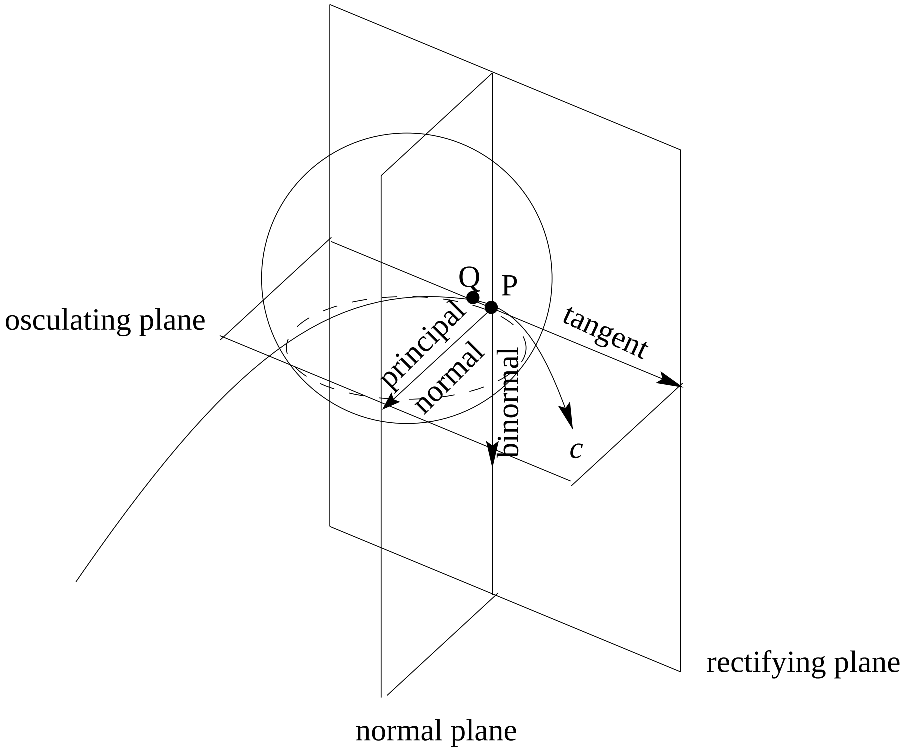


Figure 2.4: Osculating plane, normal plane and rectifying plane.

to **T** and **N**, known as the *binormal vector* **B**. The three vectors **T**, **N** and **B** form a right-handed set of mutually orthogonal unit vectors, so that $\mathbf{B} = \mathbf{T} \times \mathbf{N}$, $\mathbf{T} = \mathbf{N} \times \mathbf{B}$ and $\mathbf{N} = \mathbf{B} \times \mathbf{T}$. The planes through a given point on the curve which contain the vector **T** and **N**, **N** and **B**, and **B** and **T** respectively are known as the *osculating plane*, the *normal plane* and the *rectifying plane.* The interested reader is referred to [4] for a detailed explanation.

2.2.3 Spherical Image and Gaussian Curvature

The osculating planes of a curve may enveloped a developable surface. Developable surfaces can also be defined using the spherical image and the gaussian curvature of a surface.

Through the centre of a unit sphere we draw the diameters that are parallel to the various normals of the surface we are studying. At one point of the surface we choose one of the two directions on the normal arbitrarily and then extend this choice of a normal direction continuously to all the neighbouring points of the surface, thus obtaining a definite sense on all the normals. By choosing the same sense on the corresponding diameter of the sphere, we assign a definite point on the sphere—the end–point of the directed diameter—to every point of our surface. Thus we have a mapping of the surface onto the sphere. This process, due to Gauss, is called the *spherical representation* of the surface. The image of a surface or of a curve drawn on the sphere is referred to as the *spherical* or *Gaussian indicatrix* of the surface or of the curve. An example is shown in Figure 2.5.

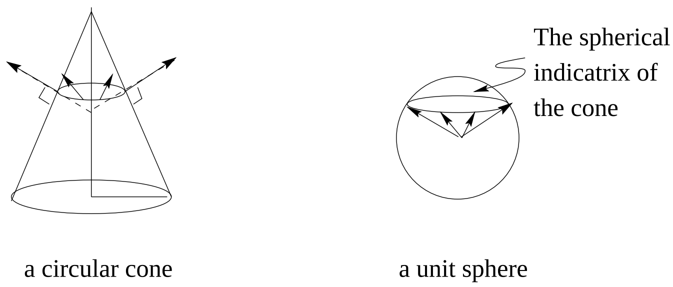


Figure 2.5: The spherical indicatrix of a cone.

Any closed curve k on the original surface is represented by a closed curve k' on the sphere. We divide the area G enclosed by k' on the sphere by the area F enclosed by k on the surface and then shrink the curve k down to a point p of the surface. As the area F approaches zero, so does the area G, and their quotient approaches a definite limit K:

$$\lim_{F \to 0} \frac{G}{F} = K.$$

The number K defined in this way is called the *Gaussian curvature* of the surface at p.

The Gaussian curvature has the important property of remaining invariant if the surface is subjected to an arbitrary bending [7]. A *bending* is defined as any deformation for which the arc lengths and angles of all curves drawn on the surface are left invariant.

Two surfaces that can be transformed into each other by bending are called "applicable" (they can be "applied") to each other.

2.2.4 Intuitive Definition of Developable Surfaces

There is a general theorem that states that a surface of constant Gaussian curvature can be transformed, by bending, into any other surface of the same constant Gaussian curvature, as Hilbert points out in [7]. It follows from this theorem that every surface whose curvature vanishes at every point can be constructed by bending a planar region. These surfaces are called *developable surfaces*.

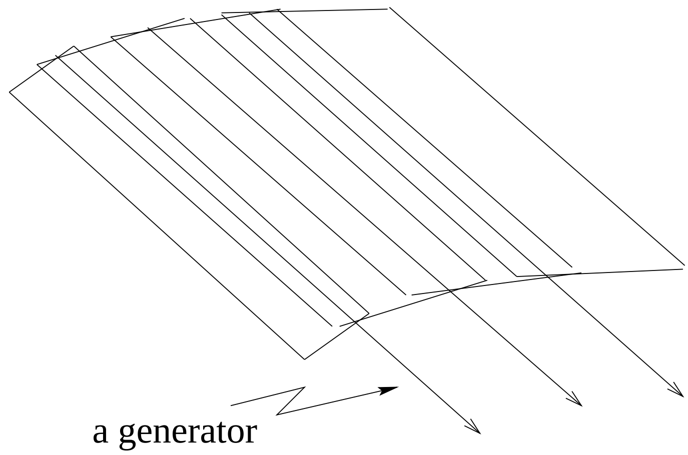


Figure 2.6: Generators of a developable surface.

Other than bending a planar region, there are two other ways of obtaining developable surfaces. Every surface enveloped by a one–parameter family of planes is a developable surface. The variable plane is tangent to such a surface along an entire straight line that is obtained as the limiting position of the line in which two neighbouring planes intersect. Since the totality of these straight lines covers the entire surface as shown in Figure 2.6, we call them *generators* of the surface [7].

Because three planes always have a point of intersection, (provided that parallel planes are regarded as planes that have an intersection at infinity,) it is plausible that any two neighbouring generators of a developable surface should have a common point. This fact leads us to third method of constructing the developable surfaces. The points of intersection of consecutive straight lines describe a curve. The generators meet the space curve tangentially. Thus, we might also define a developable surface as the surface swept out by the tangents of an arbitrary twisted curve. Such a surface can be represented using Equation 2.1 in the ruled surface definition. In this case, the twisted curve is $\alpha(t)$, and the generators are $\beta(t) = d\alpha(t)/dt$. Since

$$< \beta \times \frac{d\beta}{dt}, \frac{d\alpha}{dt} >=< \frac{d\alpha}{dt} \times \frac{d\beta}{dt}, \frac{d\alpha}{dt} >=< \frac{d\alpha}{dt} \times \frac{d\alpha}{dt}, \frac{d\beta}{dt} >= 0,$$

by Equation 2.2, the resulting surface is developable. Considered in this light, the resulting developable surface is known as the *tangential developable surface* of the curve. At the same time, the surface is also enveloped by the osculating planes of the curve. Note that for most cones and cylinders, one can not find a space curve whose tangents sweep out the cone or cylinder. Only for the cones and cylinders does this representation fail, whereas the preceding method of generation obviously applies to them as well as to the other developable surfaces. In some cases, the generators envelop a space curve at which the developable surface has a sharp edge called *edge of regression* or *cuspidal edge*, as shown in Figure 2.7. The interested reader is referred to [7] for a detailed explanation.

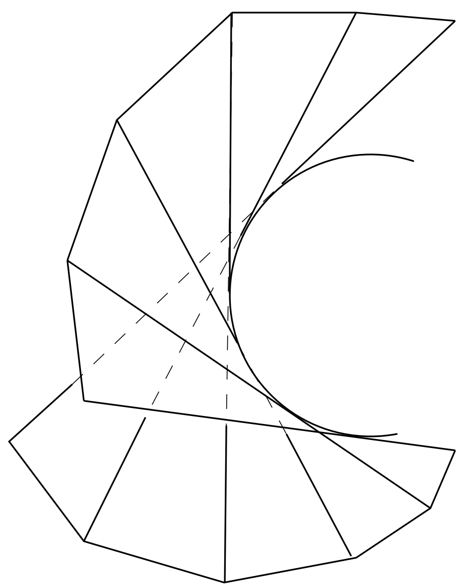


Figure 2.7: An edge of regression of a developable surface.

From the above methods of representation, one can find the spherical images of all the developable surfaces with the exception of the plane. The enveloping planes constitute the totality of planes tangent to the surfaces. Hence, all the tangent planes, and likewise all the normals, constitute a family depending on only *one* variable parameter. Therefore, the spherical indicatrix of a developable surface is always a curve. The spherical image of a surface of vanishing Gaussian curvature degenerates into a curve. It implies that the spherical image of every region of such a surface has zero area. This is to be expected from the definition of Gaussian curvature.

By the definition and intrinsic properties of developable surfaces, a developable surface can be rolled out onto a plane without stretching or tearing. In theory, the length of any curve drawn on the surface remains the same, and the area of the developable surface also remains the same [7].

2.2.5 Summary

In [9], Kergosien, Gotoda and Kunii construct developable surfaces using the fact that every surface enveloped by a one–parameter family of planes is a developable surface. They simulate the bending of a developable surface over time t by applying external forces and internal forces to the developable surface. Their system let the user specify the external forces at any time t. We will describe [9] in detail in Section 4.4.

Given a twisted curve, it is easy to compute the developable surface swept out by the tangents of the curve. However, when a user is presented with a developable surface, determining the twisted curve whose tangents sweeps out this surface is not easy.

We will make use of the properties of developable surfaces in our new developable surface representation technique, and this new technique will be presented in chapter 5. In the next chapter, we will introduce Bézier curves and surfaces in the next chapter, and they will be used in an implementation of our new technique.

Chapter 3

Bézier Curves and Bézier Surfaces

In 1962, parametric curves was developed by Bézier, an engineer at Régie Renault, for use in approximation. These techniques became the mathematical foundation of UNISURF, a design system for curves and free-form surfaces, which has been established at Renault in 1972.

We choose to use Bézier curves in our developable surfaces modelling tool. In our new modelling system, tensor product Bézier surfaces are used to approximate a particular class of developable surfaces. So in this chapter, we will briefly review the definitions and basic concepts related to Bézier curves and tensor-product Bézier surfaces. In chapter 5 and 6, we will describe how they are used in our new modelling system.

3.1 Bézier Curves

In this section, we will give a brief overview of Bézier Curves. The interested reader is referred to [6] for a detailed discussion on Bézier curves.

3.1.1 Basic Definitions

The parametric form of an n^{th} degree Bézier curve with control points $P_i = (x_i, y_i, z_i), i = 0, 1, ..., n$, is given by

$$\mathbf{p}_n(t) = \sum_{i=0}^{n} P_i \begin{pmatrix} n \\ i \end{pmatrix} t^i (1-t)^{n-i}. \tag{3.1}$$

where

$$\begin{pmatrix} n \\ i \end{pmatrix} = \frac{n!}{i!(n-i)!}, \quad \text{when } 0 \le i \le n.$$

An example of a 6^{th} degree Bézier curve is shown in Figure 3.1.

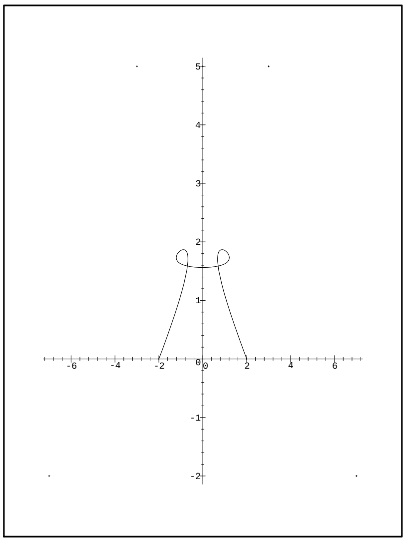


Figure 3.1: A 6^{th} degree Bézier curve.
with control points (2,0), (-3,5), (7,-2), (0,5), (-7,-2), (3,5), (-2,0).

3.1.2 Cubic Bézier Curves

The Bézier form of the cubic polynomial curve segment indirectly specifies the end-point tangent vector by specifying two intermediate points that are not on the curve. Consider a cubic Bézier curve $\mathbf{p}(t)$, $t \in [0, 1]$. An sample cubic Bézier curve is shown in Figure 3.2. The starting and ending tangent vectors are determined by the vectors

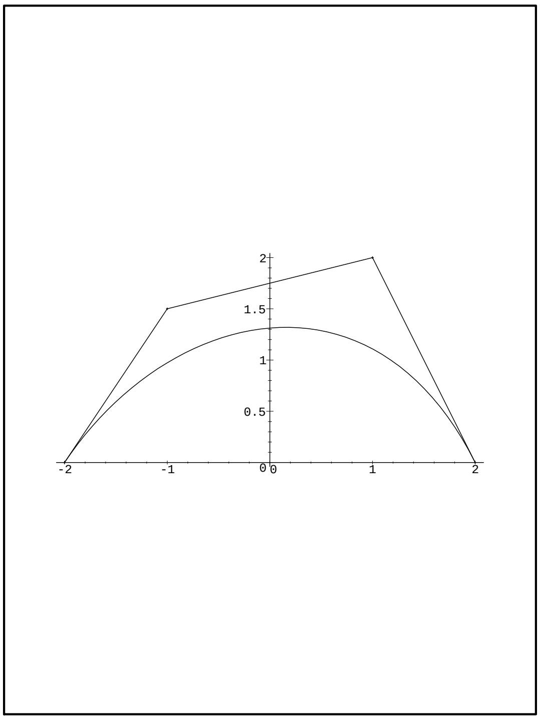


Figure 3.2: A cubic Bézier curve.

P_1P_2 and P_3P_4:

$$\frac{d\mathbf{p}}{dt}(0) = 3(P_2 - P_1), \quad \frac{d\mathbf{p}}{dt}(1) = 3(P_4 - P_3).$$

The Bézier curve interpolates the two end control points and approximates the other two.

We can specify cubic or any fixed-degree Bézier polynomials in matrix form. The Bézier geometry vector G_B, consisting of four points for a cubic, is

$$G_B = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}.$$

Observe that G_B is shortland for n column vectors, if $P_i \in \mathbb{R}^n$. In other words, G_B is a $4 \times n$ matrix. The Bézier basis matrix M_B is

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

With $T = [t^3 \quad t^2 \quad t \quad 1]$, we have

$$\begin{aligned} \mathbf{p}(t) &= TM_BG_B & (3.2) \\ &= (1-t)^3P_1 + 3t(1-t)^2P_2 + 3t^2(1-t)P_3 + t^3P_4. & (3.3) \end{aligned}$$

The four basis functions given by TM_B are the weights in the equation above, and are called the *Bernstein polynomials* of degree 3, and namely

$$B_i^3(t) = \begin{pmatrix} 3 \\ i \end{pmatrix} t^i(1-t)^{3-i}, \qquad i = 0, 1, 2, 3,$$

as a subcase of Eq 3.1. Note that for any degree n, the sum of the basis functions is unity, i.e.,

$$\sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} t^i(1-t)^{n-i} = (t + (1-t))^n = 1$$

by the binomial theorem, and each polynomial is nonnegative for $0 \le t < 1$. Thus, $\mathbf{p}(t)$ is just a moving average of the four control points. This means that each curve segment, which is just the sum of the four control points weighted by the polynomials, is completely constrained to lie in the *convex hull* of the four control points.

3.2 Tensor-product Bézier Surfaces

In this section, we will introduce tensor-product Bézier surfaces. Intuitively, a surface is the locus of a curve that is moving through space and thereby changing its shape. We can formalize this intuitive concept in order to arrive at a mathematical description of a surface. First, we assume that the moving curve is a Bézier curve of constant degree m. At any time, the moving curve is determined by a set of control points. Each original control point moves through space on a curve. Our next assumption is that this curve is also a Bézier curve, and that the curves on which the control points move are all of the same degree n.

We can parametrize the surface using the *Bernstein polynomials* $B_i^n(u)$ and $B_k^m(v)$ as separable basis functions to represent surfaces. The parametric equation

$$\mathbf{X}(u,v) = \sum_{i=0}^{n}\sum_{k=0}^{m} P_{ik} B_i^n(u) B_k^m(v) \tag{3.4}$$

defines the *tensor-product Bézier surfaces* of degree (n,m), where without loss of generality the points $u,v \in [0,1]\times[0,1]$. The coefficients P_{ik} are called the *Bézier points*, and the set of Bézier points is referred to as the *Bézier net*. Figure 3.3 shows

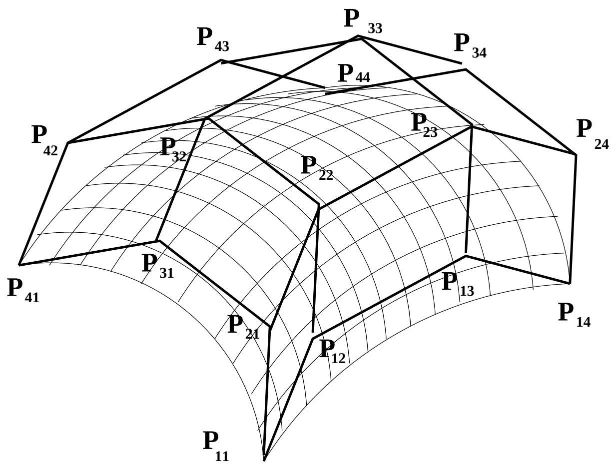


Figure 3.3: Bézier surface of degree (3,3) and its Bézier net.

a Bézier surface of degree (3,3) and its associated Bézier net. Along the parametric line $u = u_0$, the surface reduces to a isoparametric Bézier curve with Bézier points

$$P_k = \sum_{i=0}^{n} P_{ik} B_i^n(u_0).$$

Then with respect to the variable v, equation 3.4 reduces to

$$\mathbf{X}(u_0, v) = \sum_{k=0}^{m} P_k B_k^m(v).$$

The interested reader is referred to [8] for a detailed description of tensor-product Bézier surfaces.

3.3 Summary

In this chapter, Bézier curves and tensor-product Bézier surfaces are introduced. These curves and surfaces will be used in the implementation of our new technique in chapter 5. In the next chapter, we will review a number of existing algorithms for constructing developable surfaces and discuss the relationship of our new technique to these existing approaches. The concepts presented in the previous chapter will be used in the next chapter to describe some of the exisiting algorithms.

Chapter 4

Previous Work on Developable Surfaces

In this chapter, we will survey a number of interesting problems and existing algorithms for constructing developable surfaces. The four sections in this chapter introduce four main types of approaches. In the first section, a classic problem is discussed: given two distinct space curves, how to construct a continuous developable surface which connects them. In the second section, a practical problem is presented: how to build a smooth developable surface which interpolates a given set of data points. In the third section, a different problem is reviewed: given two distinct line segments x_1y_1 and x_2y_2 in $\mathbb{R}^3$ and a number of constraints, how to determine a developable surface whose four boundaries are x_1y_1, x_2y_2, a Bézier curve with end points x_1 and x_2, and another Bézier curve with end points y_1 and y_2, provided that the constraints are satisfied. In the fourth section, a simulation problem is presented: given a developable surface, how to similate the bending of that surface under external and internal forces. In the last section, we will discuss the relationship of our new technique to these existing approaches.

4.1 A Classic Problem and Its Solution

Developable surfaces are often associated with metal sheet workers. They practically solve the problem of forming a connection between two tubes of different shapes by using planar segments of metal sheets, as shown in Figure 4.1. At trade-schools and through experience, they learn how to construct such *connecting developable surfaces* (CDSs).

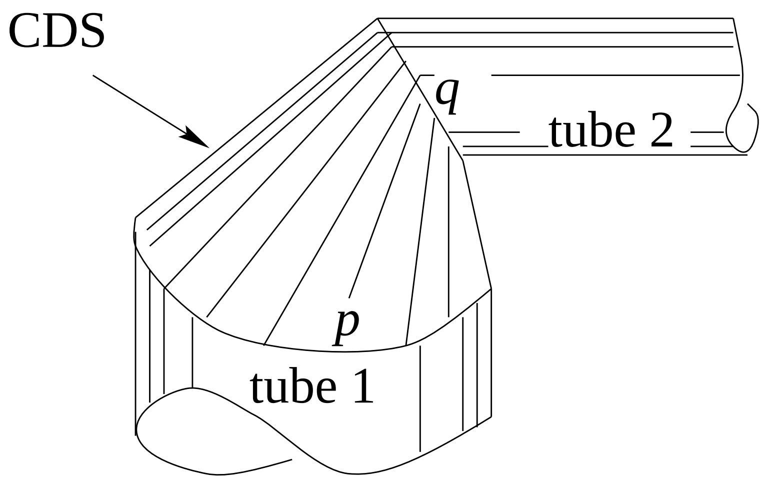


Figure 4.1: Connecting two tubes with a developable surface.

The tinsmith's job, which is to connect tube 1 and 2 by a CDS, splits into two constructive problems: First to find the CDS in 3D space, and second to bend the CDS out into the plane sheet of metal. If the boundary curve $\mathbf{p}$ of tube 1 and the boundary curve $\mathbf{q}$ of tube 2 are space-curves, the conservative designer who wants to construct the CDS on his drawing–board might find it difficult. This problem seems better adapted to computers than to classic drawing–boards.

4.1.1 Analysis

The first problem can be described as follows. Given two space curves, join the corresponding points of the two curves of forming a curved developable surface. Then flatten out the curved surface, and determine the correspondence between points on the curved surface and points on the plane sheet.

Now we will take a close look at a solution to the problem. The development is based on the fact that the curvature at point p of a curve on a developed surface is equal to the curvature of the projection of the curve onto its tangent plane of the surface at p, known as the *geodesic curvature* κ_g.

The geodesic curvature of a curve $\mathbf{r} = \mathbf{r}(u)$ is shown by Willmore (1959) to be

$$\kappa_g = \frac{\mathbf{n} \cdot (\dot{\mathbf{r}} \times \ddot{\mathbf{r}})}{\dot{s}}, \tag{4.1}$$

where $\mathbf{n}$ is the unit normal to the surface, $\mathbf{n}$, $\mathbf{r}$ and s are parametrized in terms of u.

Suppose the surface is generated by the tangent planes of two curves $\mathbf{r} = \mathbf{r}_1(u)$ and $\mathbf{r} = \mathbf{r}_2(u)$. Then for any point P with parameter u on the primary curve $\mathbf{r} = \mathbf{r}_1(u)$, the corresponding generator meets the secondary curve at a point Q with parameter u' where $[\mathbf{r}_1(u) - \mathbf{r}_2(u')] \cdot [\dot{\mathbf{r}}_1(u) \times \dot{\mathbf{r}}_2(u')] = 0$, i.e., vectors $[\mathbf{r}_1(u) - \mathbf{r}_2(u')]$, $\dot{\mathbf{r}}_1(u)$ and $\dot{\mathbf{r}}_2(u')$ are all co-planar. The equation of the surface is then given by

$$\mathbf{r} = (1 - \nu)\mathbf{r}_1(u) + \nu\mathbf{r}_2(u').$$

For example, if the curves are parametric cubics, we have

$$\mathbf{r}_i(u) = \mathbf{a}_i u^3 + \mathbf{b}_i u^2 + \mathbf{c}_i u + \mathbf{d}_i,$$

$$\dot{\mathbf{r}}_i(u) = 3\mathbf{a}_i u^2 + 2\mathbf{b}_i u + \mathbf{c}, \tag{4.2}$$

where $i = 1, 2$. The condition $[\mathbf{r}_1(u) - \mathbf{r}_2(u')] \cdot [\dot{\mathbf{r}}_1(u) \times \dot{\mathbf{r}}_2(u')] = 0$ gives a quintic equation for u' for any given u.

We first solve this equation for u', and compute the direction and length of the generator. We are then able to calculate the surface normal $\mathbf{n}$, which enables us to obtain κ_g from equation 4.1. We would then know the curvature of the developed curve, and may obtain the curve itself by integrating equation 2.6, rewritten in terms of parameter u. Since we also know the length and direction of the generator at each point P, we may obtain the development of point Q by noting that the angle between the primary curve tangent and the generator is unchanged during the development.

In a similar manner, we may locate intermediate points along the generators, so that points defined by u and v on the curved surface may be located on the development.

4.1.2 Computer Aided Construction and Development

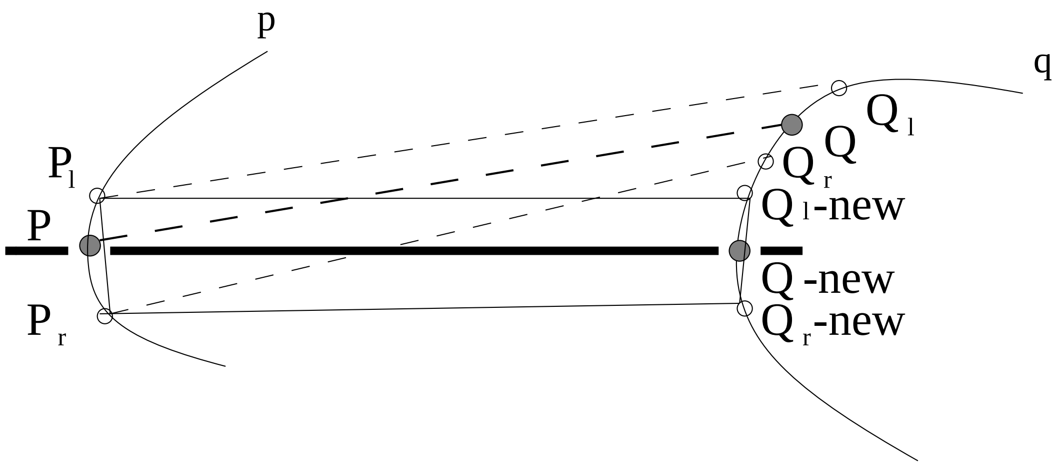


Figure 4.2: Finding the correspondence between points of two space curves.

As the surface's development in any case is based on approximative processes, it seems natural to use numerical differentiation to construct the developable surface ϕ from two given space curves. In [15], Weiss and Furtner described an algorithm which is used to find the correspondence between points of the given curves $\mathbf{p}$ and $\mathbf{q}$, as shown in 4.2:

Step 1: Parametrize the curves $\mathbf{p}$ on t_p and $\mathbf{q}$ on t_q, where $t_p \in [0,1]$ and $t_q \in [0,1]$. Initialize $t_p^* = 0$ and $t_q^* = 0$.

Step 2: Take point P on $\mathbf{p}$ and its left and right neighbour P_l, P_r on $\mathbf{p}$, where $P = \mathbf{p}(t_p^*)$, $P_l = \mathbf{p}(t_p^* + \Delta t_p)$, $P_r = \mathbf{p}(t_p^* - \Delta t_p)$, and Δt_p is some small displacement.

Step 3: Do so on $\mathbf{q}$, i.e., pick out points Q, Q_l and Q_r on $\mathbf{q}$, where $Q = \mathbf{q}(t_q^*)$, $Q_l = \mathbf{q}(t_q^* + \Delta t_q)$, $Q_r = \mathbf{q}(t_q^* - \Delta t_q)$, and Δt_q is some small displacement.

Step 4: Test. If Q_l, Q_r, P_r and P_l are coplanar to within some tolerance ϵ (i.e., span a plane), then go to step 6. Otherwise, go to step 5.

Step 5: Pick out the next triple Q-*new*, Q_l-*new* and Q_r-*new* on q, adjust t_q^*, and go to step 4. Note that if we have the right Q to a certain P, the determinant D of the vector P_lP_r, Q_lQ_r and P_lQ_r would vanish. We can discretize $\mathbf{q}$. If for two consecutive parameter values t_q-*old* and t_q-*new* the determinants D-*old* and D-*new* have different signs, the right Q would correspond to a value of t_q that is between t_q-*old* and t_q-*new* [15].

Step 6: Store the Coordinates of P and Q and the counting index of the generator PQ.

Step 7: $t_p^* := t_p^* +$ some step size. Go on with the next P on $\mathbf{p}$, repeat step 1–7, and so on until finished.

Given two space curves, one can use this algorithm to join the corresponding points of the two curves to form a curved developable surface.

4.2 Developable Surface Synthesis Using Geodesics

In this section, we will consider a different problem. Assume that the indicatrix N of a developable surface D is approximated by $n+1$ points on the unit sphere and that those points are the spherical image of corresponding points of the geodesic X, as shown in Figure 4.3. As we shall see later, the geodesic X will be used to construct the developable surface. Also, assume that the orientations of the tangent plane and the arc-length along the geodesic X at those discrete points are known. In [11], Redont describes an algorithm which produces a discrete representation for a developable surface based on the above information. Since $n+1$ points on the indicatrix N are known, the algorithm approximates the indicatrix N by a piecewise continuous curve C which interpolates the $n+1$ points. Then the arc-length information is used to construct a family of circular cones, each with a geodesic segment which corresponds to one segment of the original geodesic. The desired developable surface is constructed using patches of the circular cones. Next, the techniques used in this algorithm will be presented.

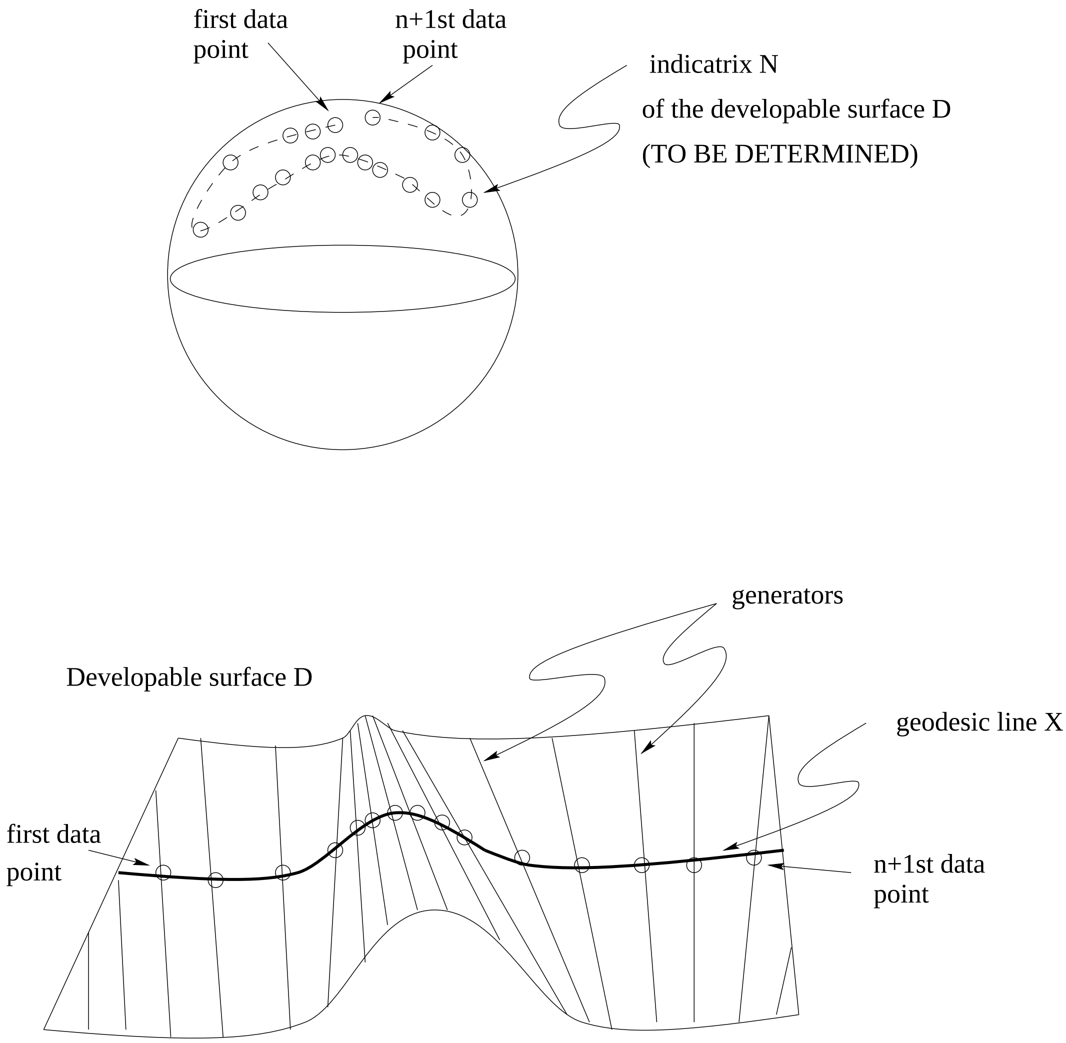


Figure 4.3: Developable surface D, its geodesic X and its spherical indicatrix N.

4.2.1 Approximating the Spherical Indicatrix

First, we consider $n+1$ points $M_0, M_2, ..., M_{2n}$ [1] on the unit sphere together with tangent vectors $t_0, t_2, ..., t_{2n}$, which are meant to build a discretized spherical indicatrix of developable surface D. If the tangent t_{2i} is not part of the original data they can be estimated in various ways from points M_{2i}. Redont shows in [11] that one can find an arc-of-circle interpolation of points M_{2i} on the sphere which respects tangents t_{2i} using the stereographic projection.

[1] Only for convenience are the indices of the points supposed to be even. We will add an intermediate point M_{2i+1} between points M_{2i} and M_{2i+2}, for $i = 0, ..., n-1$, to approximate the spherical indicatrix.

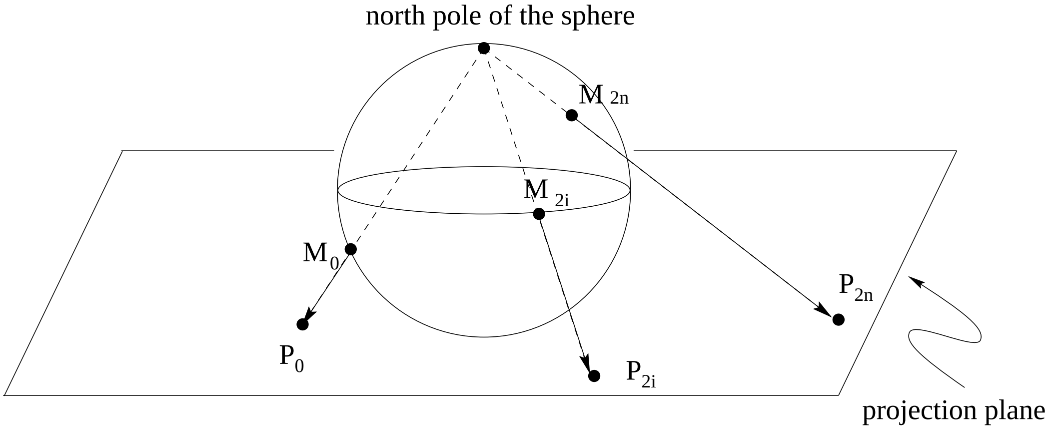


Figure 4.4: The stereographic projection.

Consider a sphere and the horizontal plane tangent to its south pole, as shown in Figure 4.4. The stereographic projection maps the sphere, deprived of its north pole, onto the plane. The image of a point on the sphere is the intersection between the plane and the straight line defined by the point and the north pole. The stereographic projection possesses the interesting quality of preserving angles and circles [7].

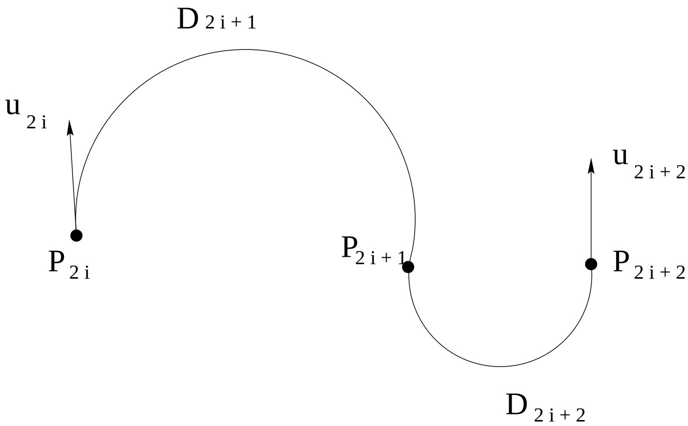


Figure 4.5: Interpolating points P_{2i} and P_{2i+2} using two arcs of circles.

The interpolation problem on the sphere can be reduced to the same problem in the plane. Let P_{2i} and u_{2i} be the stereographic images of M_{2j} and t_{2i}. Given two points P_{2i} and P_{2i+2} with tangents u_{2i} and u_{2i+2}, points P_{2i} and P_{2i+2} can be connected with two arcs of circles D_{2i+1} and D_{2i+2} meeting tangentially at, say, point P_{2i+1} such that D_{2i+1} is tangent to u_{2i} at point P_{2i} and D_{2i+2} is tangent to u_{2i+2} at point P_{2i+2}. Then let points M_{2i+1} and arcs C_i be the images under the inverse stereographic projection of points P_{2i+1} and of arcs D_i as shown in Figure 4.5. The curve C formed by arcs of circles C_i interpolates the initial points M_{2i} and tangents t_{2i} on the unit sphere.

4.2.2 Approximating the Spherical Mapping

From the previous step, we have built a curve C that approximates the spherical indicatrix N. The spherical mapping assigns some point N_σ on the spherical indicatrix to point $X(s)$ on the geodesic, where σ is the arc length of N and s is the arc-length of X. An approximation to the spherical mapping as a function of arc-length s is required, such that the image of X is curve C. This is achieved by defining the arc-length τ of C as a function of s in such a way that point $C(\tau)$ lies near point $N(\sigma)$. Here, τ is defined piecewise on each arc C_i as a function of the form $\tau_i - \cot(\psi_i)\arctan(k_i s + l_i)$, where τ_i, k_i and l_i are unknown constants, and $\cos\psi_i$ is the radius of arc of circle C_i. We can construct cones K_i using arcs of circles C_i, and let ψ_i be the angle between the axis of cone K_i and any of its generators. The constants τ_i, k_i and l_i can be determined by some formulae. That gives us τ as a function of s. Thus C together with its parametrization by $\tau(s)$ defines an approximation to the spherical mapping. Then the aim is to assemble cones K_i and K_{i+1} along a common generator to build a smooth surface K with one particular geodesic Ξ consisting of segments Ξ_i which is drawn on cone K_i. It is expected that curve Ξ will be differentiable at the joint between K_i and K_{i+1}, so it is the angles of Ξ_i and Ξ_{i+1} with the generator common to K_i and K_{i+1} are equal

$$arccotan(k_i s_i + l_i) = arccotan(k_{i+1} s_i + l_{i+1}), i = 1, ..., 2n-1. \tag{4.3}$$

Given that the angle between geodesic X of surface D and the generator crossing X at the point of arc-length s_0 is $a(s_0) = a_0$, we can use k_i and $\tau(s)$ along with 4.3 to determine how the geodesic Ξ_i is drawn on cone K_i. Thus $2n$ patches of circular cones K_i are defined, and they are bounded by generators and with a segment of one particular geodesic Ξ_i is drawn on cone K_i. The interested reader is referred to [11] for a detailed explanation.

4.2.3 Representing the Developable Surfaces

At the end of the previous steps, a family of circular cones K_i, $i = 1, ..., 2n$, has been constructed, each with a distinguished geodesic segment Ξ_i. Each cone, however, is defined up to a translation. If the apex of cone K_i coincides with the origin, the geodesic segment Ξ_i is given by

$$\Xi_i(s) = \frac{1}{k_i}(1 + (k_i s + l_i)^2)^{1/2}\Gamma(\tau(s)),$$

where $s_{i-1} \leq s \leq s_i$ and $\Gamma(\tau)$ is the unit geodesic normal to curve C. Let each cone K_{i+1} be translated so that points $\Xi_i(s_i)$ and $\Xi_{i+1}(s_i)$ coincide. Then two consecutive cones K_i and K_{i+1} have the same generator through point $\Xi_i(s_i)$, the direction of which is given by the geodesic normal $\Xi(\tau(s_i))$ to curve C at point $M_i = C(\tau(s_i))$. Cones K_i and K_{i+1} have the same tangent plane along their common generator, the direction of which is given by the plane tangent to the unit sphere at point M_i. A smooth discrete representation for our initial developable surface D has been obtained.

4.3 Interpolation with Developable Bézier Patches

In this section, we will outline the idea of a unique approach which is discussed in [1]. Though it is not closely related to the new technique which we present in this thesis, which also uses Bézier patches.

The problem is defined as follows. Given the end points of an m^{th} degree Bézier curve parametrized by $\mathbf{x}_A(t)$ and the end points of an n^{th} degree Bézier curve parameterized by $\mathbf{x}_B(t)$, $t \in [0, 1]$, the family of lines defined by $\mathbf{x}_A(t)\mathbf{x}_B(t)$ for all $t \in [0, 1]$ forms a surface. We would like to know how the control points of $\mathbf{x}_A(t)$ and $\mathbf{x}_B(t)$ should be related, so that the resulting surface is a developable surface, provided that some constraints are satisfied.

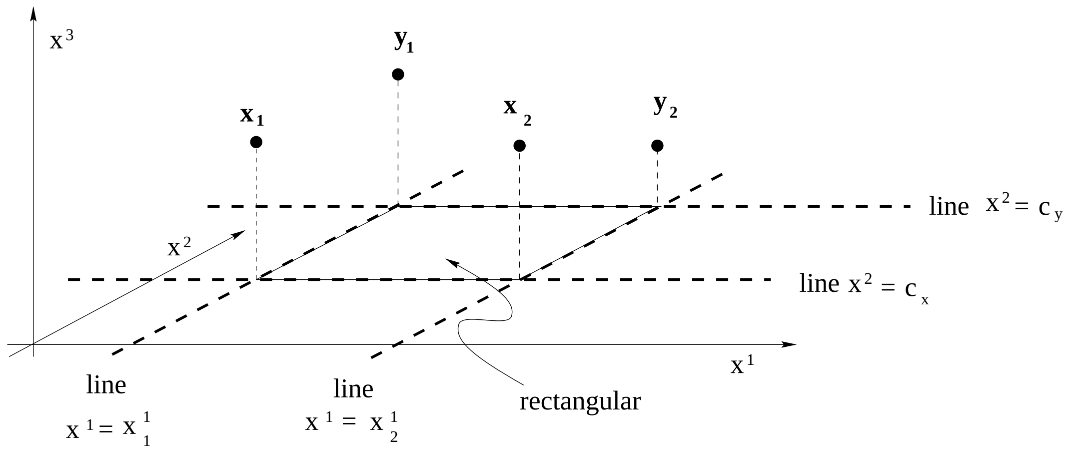


Figure 4.6: The rectangular grid.

4.3.1 Basic Concepts

Note that we will use the same notation as in [1]. Denote the coordinates of a point $\mathbf{a}$ by a^i, $i = 1, 2, 3$, so that $\mathbf{a} = (a^1, a^2, a^3)$. Let $\mathbf{x_1}(x_1^1, c_x, x_1^3)$, $\mathbf{x_2}(x_2^1, c_x, x_2^3)$, $\mathbf{y_1}(x_1^1|c_y|y_1^3)$, $\mathbf{y_2}(x_2^1|c_y|y_2^3)$ be four points (over a rectangle in the ground plane $x^3 = 0$), as shown in Figure 4.6. Further, let

$$C_A : \mathbf{x_A}(t) = \sum_{i=0}^{m} \mathbf{a_i} B_i^m(t), \quad t \in [0, 1],$$

and

$$C_B : \mathbf{x_B}(t) = \sum_{i=0}^{n} \mathbf{b_i} B_i^n(t), \quad t \in [0, 1],$$

be parametric Bézier curves of degree m and n $(n \geq m)$ in the planes $x^2 = c_x$ and $x^2 = c_y$, respectively, with

$$\mathbf{a_0} = \mathbf{x_1}, \quad \mathbf{a_m} = \mathbf{x_2}, \quad \mathbf{b_0} = \mathbf{y_1}, \quad \mathbf{b_n} = \mathbf{y_2},$$

$$\mathbf{a_0}, \mathbf{a_1}, ..., \mathbf{a_m} \text{ lie in plane } x^2 = c_x, \text{ and } \mathbf{b_0}, \mathbf{b_1}, ..., \mathbf{b_n} \text{ lie in plane } x^2 = c_y,$$

and where

$$B_i^k(t) = \begin{pmatrix} k \\ i \end{pmatrix} t^i (1 - t)^{k-i}.$$

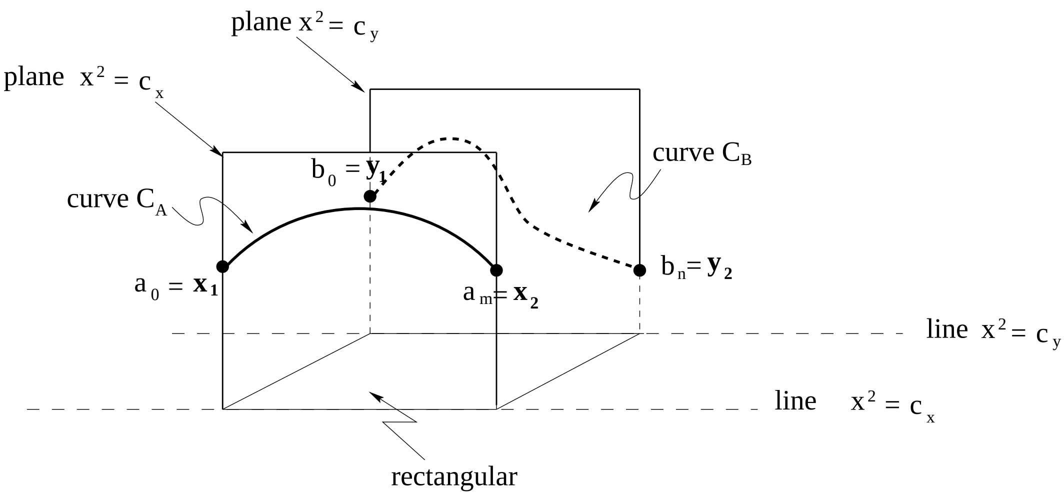


Figure 4.7: The parallel planes.

The configuration is as shown in Figure 4.7. Then we call a developable Bézier patch

$$\Phi : \quad \mathbf{X}(t,u) = (1-u)\mathbf{x_A}(t) + u\mathbf{x_B}(t), \quad (t,u) \in [0,1] \times [0,1]$$

an *interpolating developable* m, n*-patch* or an *id-patch* with the design curve c_A. The curve C_B will be determined from the curve C_A using the properties of developable surfaces. The id-patch is called *admissible* iff Φ does not contain singular points.

In [1], Aumann deals with interpolating developable 3,4-patches. Let curve c_A be a Bézier curve of degree three with control points

$$\mathbf{x}_1 = \mathbf{a}_0, \quad \mathbf{a}_1, \quad \mathbf{a}_2, \quad \mathbf{a}_3 = \mathbf{x}_2,$$

where

$$a_0^1 < a_1^1, \quad a_2^1 < a_3^1, \quad a_1^2 = a_2^2 = c_x.$$

Let the fourth degree Bézier curve c_B have control points

$$\mathbf{y}_1 = \mathbf{b}_0, \; \mathbf{b}_1, \; \mathbf{b}_2, \; \mathbf{b}_3, \; \mathbf{b}_4 = \mathbf{y}_2,$$

where

$$b_1^2 = b_2^2 = b_3^2 = c_y.$$

An example id-patch is shown in Figure 4.8.

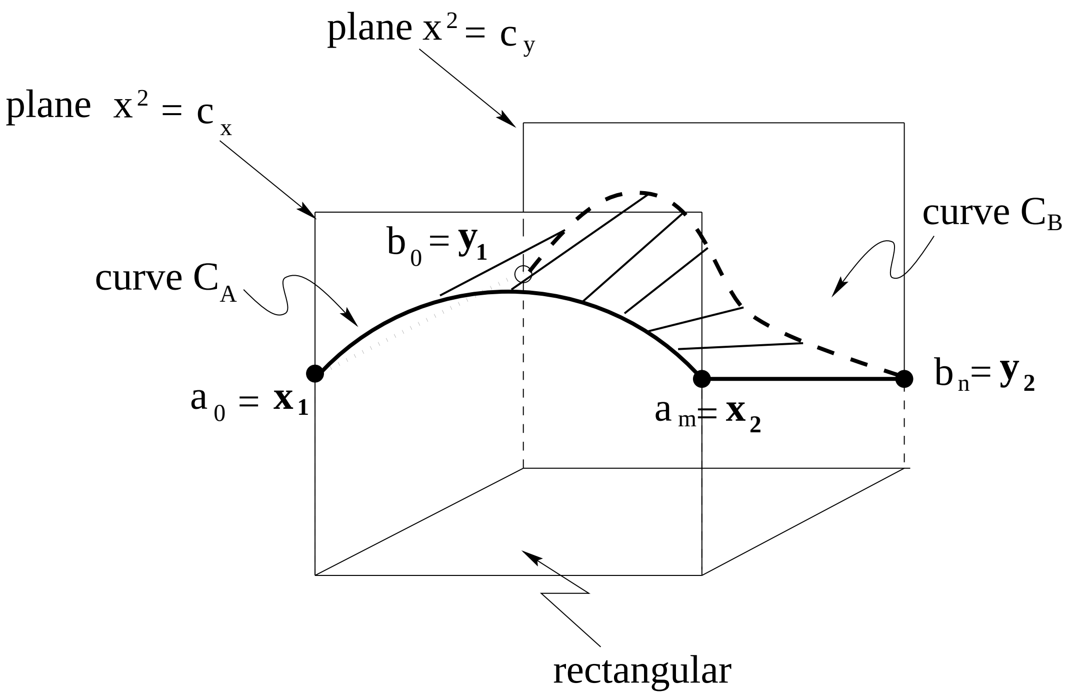


Figure 4.8: An ID-patch.

Since the tangent plane, at a point of the ruling $t = t_0$, contains the direction of the ruling, the patch Φ is developable iff for all $t \in [0, 1]$, $\mathbf{x}_A(t) - \mathbf{x}_B(t)$, $\dot{\mathbf{x}}_{\mathbf{A}}(t)$, $\mathbf{x_A}(t) - \mathbf{x_B}(t)$ and $\dot{\mathbf{x}}_{\mathbf{B}}(t)$ are coplanar. Since the projection of the id-patch on the x^1x^2-plane is rectangular by the definition of id-patches, this condition is equivalent to

$$\dot{\mathbf{x}}_A(t)||\dot{\mathbf{x}}_B(t), \quad \forall t \in I$$

or

$$\dot{\mathbf{x}}_{\mathbf{B}}(t) = \rho(t)\dot{\mathbf{x}}_A(t),$$

for a suitable ρ. Since c_A is a 3^{rd} degree Bézier curve and c_B is a 4^{th} degree Bézier curve, the function $\rho(t)$ can be written as

$$\rho(t) = kt + h, \quad t \in [0, 1].$$

The objective of the analysis is to determine possible coefficients k and h of $\rho(t)$ which lead to developable id-patches that satisfy given constraints [1].

4.3.2 Results

In [1], theorems are given which answer the following questions:

- How do the Bézier points $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ depend on the design curve c_A?
- How can the Bézier points $\mathbf{a}_1, \mathbf{a}_2$ be chosen to get an admissible patch?
- How can the id-patches be connected so that it satisfies continuity constraints under given conditions?

At present, researchers can draw conclusions about the design criteria of a small subclass of developable surfaces. For instance, in [1] Aumann deals with the subclass of developable Bézier patches whose projections on the x^1x^2–plane are rectangular according to the basic condition in his initial definition of $id-patches$. Not very many developable surfaces which we see in everyday life can be represented by id-patches. Hence, it is difficult to construct general developable surfaces using this approach.

4.4 Simulating the Bending of Developable Surfaces

The approaches which we have presented so far do not deal with the bending of the developable surface under external and internal forces. In this section, we will introduce a mathematical model for simulating the bending of developable surfaces under those forces. This model appears in [9].

The technique deals with C^2 continuous developable surfaces. Such surfaces do not have edges of regression. The developable surface is a plane, or it can be generated as the envelope of a single regular one-parameter family of planes. The system let the user specify the forces exerted on a developable surface. Then the system let the surface respond to the external and internal forces and produce a bending which preserves the surface metrics.

We will present the model in two parts: the representation of developable surfaces and the model for bending.

4.4.1 Representing Developable Surfaces

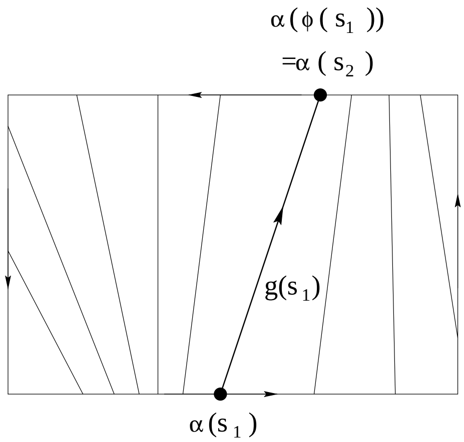


Figure 4.9: The correspondence function $\phi(s)$.

Let S_0 be a convex plane region and let S_t be a nonplanar developable surface isometric to S_0 at time t. Assume that the boundaries of S_0 and S_t are piecewise differentiable curves α_0 and α_t, respectively, parametrized by arc-length s. S_t is swept by its family of generators, and each generator crosses the boundary of S_t at two points, except at the two ends [9]. Each point $\alpha_t(s_1)$ on the boundary can be mapped onto the point at the other extremity of its generator, say $f(\alpha_t(s_1))$. There is some unique s_2 for which $\alpha_t(s_2)) = f(\alpha_t(s_1))$, so we can define the map $\phi(s_1) = s_2$, as shown in Figure 4.9.

Let $g_t(s)$ be the generator of S_t passing through $\alpha_t(s)$, and let $g_0(s)$ be the image of $g_t(s)$ when S_t is developed onto S_0, as shown in Figure 4.10, namely,

$$g_t(s) = \alpha_t(\phi_t(s)) - \alpha_t(s)$$

and

$$g_0(s) = \alpha_0(\phi_t(s)) - \alpha_0(s).$$

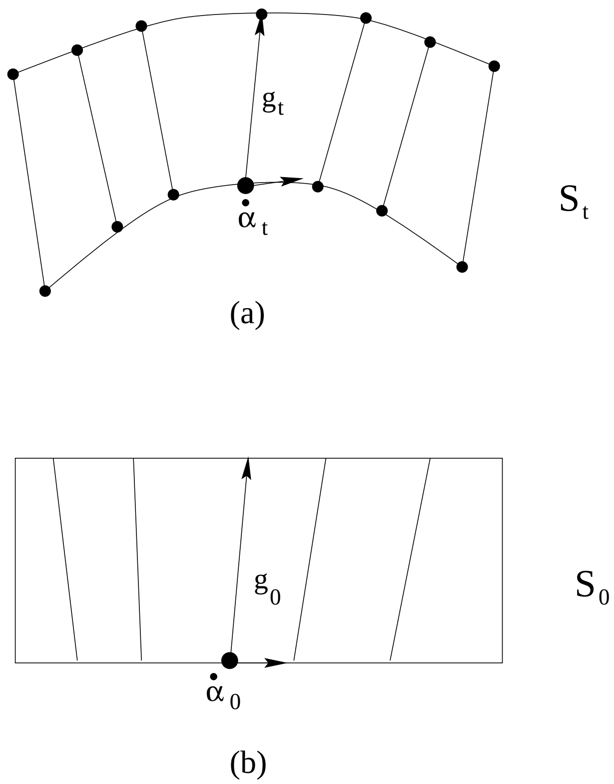


Figure 4.10: Applying the developable surface S_t to the planar surface S_0.

Since surface S_0 and S_t have the smae metric, they are related by

$$\begin{cases} < \dot{\alpha}_t(s), \dot{\alpha}_t(s) >=< \dot{\alpha}_0(s), \dot{\alpha}_0(s) >= 1 \\ < \dot{\alpha}_t(s), g_t(s) > - < \dot{\alpha}_0(s), g_0(s) >= 0 \\ < g_t(s), g_t(s) > - < g_0(s), g_0(s) >= 0 \end{cases}, \tag{4.4}$$

where $< x, y >$ denotes the inner product of vectors x and y in $\mathbb{R}^3$ and $\dot{\alpha} = d\alpha/ds$. We represent S_t by the pair of functions $(\alpha_t(s), \phi_t(s))$. By Equation 4.4, the surface has the same tangent plane at $\alpha(s)$ and $\alpha(\phi(s))$, for any s. The developable surface is the envelope of that family of common tangent planes. In the next subsection, we will use this developable surface representation to describe a model for bending.

4.4.2 Modelling the Bending of Developable Surfaces

We discretize $\alpha_t(s)$ by n nodes, each of which is connected to its neighbouring nodes by line segments as shown in Figure 4.10(a). The bending of a developable surface will be simulated over time t. At any time t, external forces can be specified by the user, and internal forces are computed by the modelling system using simple formulas [9]. Let $F_i(t) \in \mathbb{R}^3$ be the net force exerted on node i at time t. The forces which are applied to the nodes will attempt to displace the nodes, and only few of the displacement combinations would preserve the surface metrics. In other words, we want to look for the small displacement $(d\alpha_t(s), d\phi_t(s))$ due to forces $F_i(t)$, $i = 0, ..., n-1$, where the resulting surface represented by the pair of functions $(\alpha_t(s) + d\alpha_t(s), \phi_t(s) + d\phi_t(s))$ preserves the surface metrics.

Let X be the function space consisting of all pairs of functions $(\alpha(s), \phi(s))$. Let Y be its subspace where each element of Y satisfies the constraints in Equation 4.4. The surface S_t at time t is represented by a point y_t in Y and the evolution of S_t under bending is represented by a curve in Y passing through y_t as shown in Figure 4.11. If

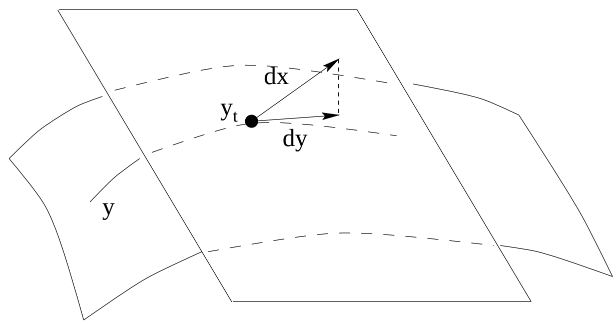


Figure 4.11: Projection method.
The surface represents the function space Y.
The displacement dx is projected onto the tangent vector dy.

we add an infinitesimally small $d\alpha_t(s)$ to $\alpha_t(s)$, y_t moves in the direction $(d\alpha_t(s), 0)$. If this dx is contained in the linear space TY tangent to Y at y_t, then $y_t + dx$ will still be in Y, and we get true bending. If dx is not in the tangent linear space, the surface represented by the pair of functions $(\alpha_t(s) + d\alpha_t(s), \phi_t(s))$ violates the constraints in Equation 4.4. We thus project dx onto the tangent linear space tangent to the space of constrained positions. It is a classical fact that the projected component dy is also the permitted displacement that best approximates dx in TY [9].

The equation of the tangent linear space at y_t can be derived from Equation 4.4. Note that if dv and dw represent small variations of vector v and w, respectively, the variation of an inner product $< v, w >$ is $< dv, w > + < v, dw >$. From Equation 4.4, we know that any variation $dy_t = (d\alpha_t(s), d\phi_t(s))$ in the tangent linear space should satisfy the system of linear equations

$$\begin{cases} < d\dot{\alpha}_t(s), \dot{\alpha}(s) >= 0 \\ < d\dot{\alpha}_t(s), g_t(s) > + < \dot{\alpha}_t(s), dg_t(s) > - < \dot{\alpha}_0(s), dg_0(s) >= 0 \\ < dg_t(s), g_t(s) > - < dg_0(s), g_0(s) >= 0 \end{cases} , \tag{4.5}$$

where

$$dg_t(s) = d\alpha_t(\phi_t(s)) + \dot{\alpha}_t(\phi_t(s))d\phi_t(s) - d\alpha_t(s)$$

and

$$dg_0(s) = \dot{\alpha}_t(\phi_t(s))d\phi_t(s).$$

The bending of a developable surface can be calculated using Equations 4.4 and 4.5.

In [9], finite bendings are computed by numerically integrating the field of the projected vectors using a Runge-Kutta method, and the constraints are restored periodically using a Newton's method.

Such an approach or a similar approach might be used in an extension of our new system to animate the movement of a developable surface under some forces, but this is beyond the scope of this thesis. The focus of this thesis is the representation of developable surfaces.

The examples presented in [9] deal with the creasing of applicable surfaces as well as bending of developable surfaces. The metric constraint is slightly relaxed, and the surfaces are allowed to be stretched slightly. Since we are not concerned with applicable surface in this thesis, we will not present algorithms for simulating the creasing of applicable surfaces.

4.5 Discussion

In this section, we will briefly describe our new approach and discuss the relationship between the new approach and the existing techniques.

Our new technique takes advantage of the geometric characteristics of the surface to be modelled. First, the surface to be modelled is divided into several pieces. Then the shape of each surface piece is defined by a generalized cone, where the shape of the generalized cone is defined by its cross section together with its apex. The cross section of the cone can be designed interactively using piecewise continuous curves. Each 3D surface piece is constructed separately. Then surface pieces are stitched together to construct the desired developable surface.

The idea of dividing the surface into pieces and approximating each piece separately is also used in the approach discussed in Section 4.2. In that approach, a circular cone is used to approximate the shape of each surface piece. In our approach, a generalized cone is used to define the shape of each surface piece.

To define the shape of a generalized cone, we allow a user to specify the apex position and a planar cross section of the cone. This is a special case of the classic problem we talked about in Section 1 of this chapter. In the classic problem, given two space curves, we join the corresponding points of the two curves to form a curved developable surface. To define a generalized cone, we let one of the curves be a point and let the other curve be a planar curve.

Our new technique combined a number of ideas from the existing techniques mentioned in Section 1 and 2. However, it is very different from those approaches. For instance, in the *Geodesic* approach in Section 4.2, the surface is subdivided into patches based on the locations of input sampling points along the geodesic. To get a better approximation of the desired shape, the user has to input more sampling data points, therefore the surface is subdivided into smaller patches. In our new technique, we divide the surface into several pieces based on its appearance and geometric features. We can adjust the shape of each surface piece locally to obtain a better approximation of the desired developable surface. It is not necessary to subdivide each surface piece into smaller and smaller surface patches.

In the next chapter, we will describe our developable surface modelling technique in details.

Chapter 5

A New Modelling Primitive

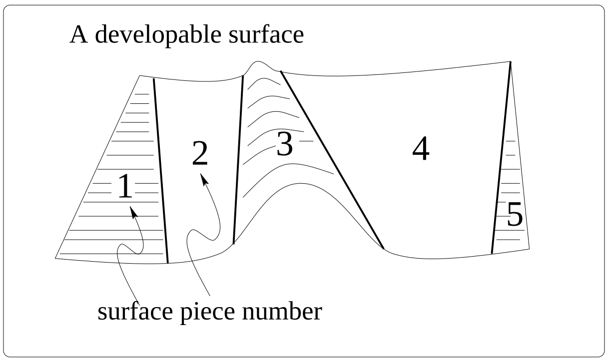
In this chapter, we present a new modelling system that can be used to create developable surfaces from scratch. In the first section, we discuss the main idea. In the second section, we describe the structure of the system. We will show how the modelling of a developable surface is broken into sub-tasks of modelling surface pieces individually. In the third section, we explain how the sub-tasks and the main task are accomplished.

5.1 Main Idea

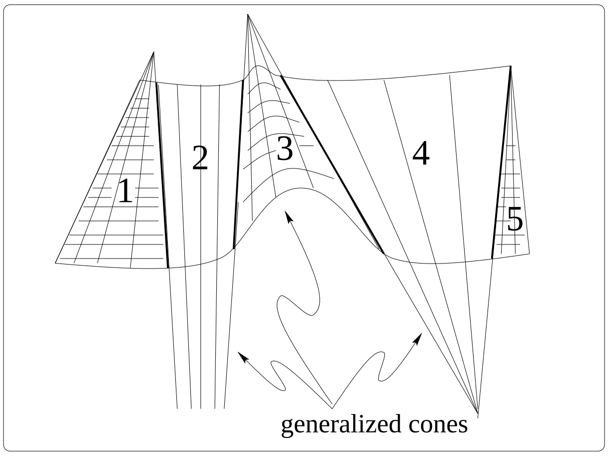
We divide the surface into several pieces based on the geometry of the surface as shown in Figure 5.1(a). Note that currently our modelling tool is set up to deal with sequential developable surface pieces only, and it handles sheets of polygonal shape. We approximate each piece by a generalized cone as shown in Figure 5.1(b). To define a generalized cone, we specify a cross section and the position of the apex in relation to this cross section as shown in Figure 5.1(c).

5.2 Data Structure

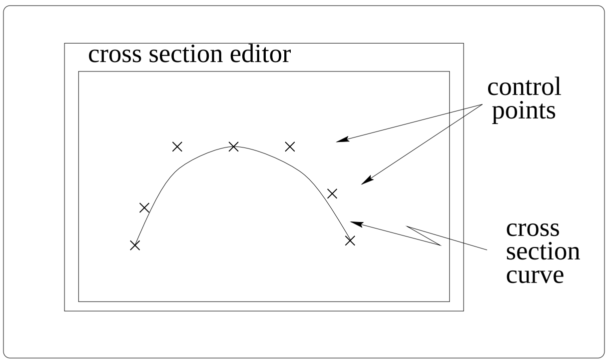
The data structure of the model has four levels as shown in Figure 5.2(a). To obtain the desired developable surface, we do the construction from bottom-up.



(a) Divide the surface into a number of pieces.



(b) Define the shape of each surface piece
by a generalized cone.



(c) Use piecewise continuous Bézier curves
to define a cross section of a generalized cone.

Figure 5.1: Dividing the developable surface into pieces and define each surface piece by a generalized cone.

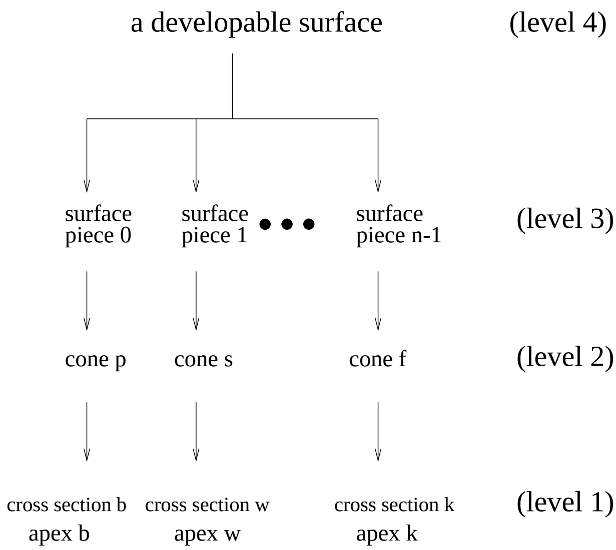


Figure 5.2: Data structure.

On level–4, a developable surface is desired.

On level–3, the developable surface is divided into n surface pieces.

On level–2, each surface piece is associated with a generalized cone. The shape of a surface piece can be determined by the shape of the corresponding cone and some simple initial conditions. Several surface pieces can be associated with the same generalized cone, possibly, with different initial conditions. For instance, if the shapes of two ribbon pieces are approximately the same, then they may use the same generalized cone to define their shapes.

On level–1, each cone is defined by a cross section and an apex.

5.3 Implementation Issues

As shown in Figure 5.2, the main task is broken into smaller tasks from the top (level–4) to the bottom (level–1).

5.3.1 Step 1: Cross Section → Cone

User input occurs only on the lowest level, namely level 1. On this level, users have already decided upon the subdivision of the surface. For each surface piece, the user specifies a cross section, an apex and a few initial conditions.[1] The initial conditions will determine the relative position of the apex and the cross section and the region that needs to be trimmed from the generalized cone to obtain the surface block. Each cone is defined in its own 3D local coordinate system.

5.3.2 Step 2: Mapping Between Cones and Surface Pieces in 2D

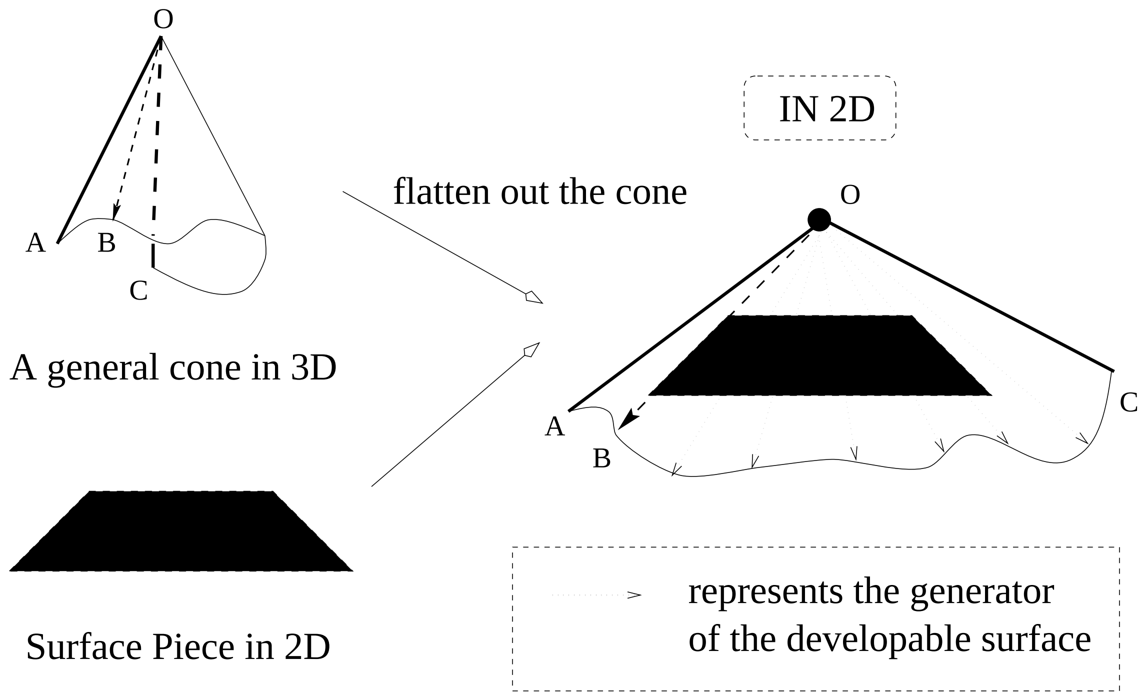


Figure 5.3: Mapping between a cone and a surface piece in 2D.

As shown in Figure 5.3, when a cone is flattened out, for a given point on the surface piece, we can easily locate the generator passing through it. Clearly, when the cone is flattened out, the user-specified cross section corresponds to a plane curve. When

[1]The cross section can be interactively designed as shown in Figure 5.1(c), or its control points can be specified in a script file.

we are dealing with a generalized cone, it is difficult to write down a closed form formula for this plane curve. However, we may use a system of differential equations and solve the problem numerically. We can calculate the surface profile by making the following observations.

- **Observation 1:** The curvature $\kappa(t)$, the angular velocity $\dot{\psi}(t)$ and the speed $\dot{s}(t)$ of this plane curve are connected by the simple relation

$$\dot{\psi}(t) = \dot{s}(t)\kappa(t).$$

  Proof: By equation 2.3

$$\kappa = \frac{d\psi}{ds} = \frac{d\psi}{dt}\frac{dt}{ds} = \frac{\dot{\psi}}{\dot{s}}.$$

$\square$

- **Observation 2**: The velocity at a point $(x(t), y(t))$ on the plane curve is described by the equation

$$\dot{x}(t) = \dot{s}(t)\cos\psi(t)$$

$$\dot{y}(t) = \dot{s}(t)\sin\psi(t).$$

  Proof: By 2.4 and 2.5,

$$\frac{dx}{ds} = \cos\psi$$

  and

$$\frac{dy}{ds} = \sin\psi.$$

  Multiplying both sides of each equation by $\frac{ds}{dt}$, assuming $\frac{ds}{dt} \neq 0$, we obtain

$$\frac{dx}{dt} = \frac{ds}{dt}\cos\psi$$

  and

$$\frac{dy}{dt} = \frac{ds}{dt}\sin\psi.$$

$\square$

Therefore, we can determine the profile of this plane curve using the system of differential equations

$$\begin{bmatrix} \dot{\psi}(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \dot{s}(t)\kappa(t) \\ \dot{s}(t)\cos\psi(t) \\ \dot{s}(t)\sin\psi(t) \end{bmatrix}$$

along with the initial values

$$\begin{bmatrix} \psi(0) \\ x(0) \\ y(0) \end{bmatrix} = \begin{bmatrix} \psi_0 \\ x_0 \\ y_0 \end{bmatrix}.$$

Note that $\kappa(t)$ can be obtained using equation 4.1 and $\kappa(t) = \kappa_g$. As mentioned earlier, the curvature at point p of a curve on the developed surface is equal to the curvature of the projection of the original curve onto the tangent plane of the surface at p.

5.3.3 Step 3: Construct the 3D Developable Surface

Using the 2D mapping shown in Figure 5.3, we can find, for each point on the surface piece, its counterpart on the cone. Since each cone is defined in its own 3D coordinate system, we can represent the ribbon piece in this 3D local coordinate system as well.

Next, we need to position the surface piece in the 3D world coordinate system. Currently, the system is set up to deal with sequential developable surface pieces. Let the developable surface pieces be numbered $0, 1, 2, ..., n-1$, and the last surface generator of the i^{th} surface piece is the first surface generator of the $(i+1)^{st}$ surface piece.

To properly connect the two adjacent surface pieces, we want the shared surface generator to be correctly aligned and the surface normals of each piece at that boundary to be parallel. A unique linear transformation matrix can be determined using these constraints, so that the surface piece $i+1$ is transformed from its local coordinate system into the world coordinate system. The output of this step is the resulting developable surface as a whole in 3D.

This step is automatically carried out by the system, and no user input is required at this step.[2]

5.3.4 G^1 Continuity of the Developable Surface

Since we use piecewise C^1 continuous cubic Bézier curves to design the cross section of a cone associated to a surface piece, clearly, each surface piece is C^1 continuous. By the surface construction process described above, two adjacent surface pieces join each other at a common generator, and the surface normals of the two surface pieces along this generator are the parallel. In other words, the adjacent surface pieces have the same tangent plane along their shared generator. Hence, the surface normal along this boundary is G^1 continuous. That implies that the developable surface is also G^1 continuous across the boundary of two adjacent surface piece. Therefore, the developable surface constructed using our method is G^1 continuous.

5.3.5 Discussion

The above steps are used for generalized cones. In some simple cases, we will be able to write down the explicit equation of the edge of the developable surface.

Now we consider a polygon in 2D with vertices $ABCD$ as shown in Figure 5.4. Note that $AB||CD$ and $\angle ADC = \angle BCD = \alpha$. In the configuration in 3D, the polygon wraps around the circular cone in such a way that the AB and DC coincide along the straight line OC. Clearly, OC is a generator of the cone. The circular cone is specified by the apex O and the circular cross section centred at O' with radius r, where OO' is perpendicular to the circular cross section at O' and the distance from O to any point on the boundary of that circle is l.

We can parametrize the position of the point P along the edge AB in 3D with respect to the angle θ, where θ is the angle between OP and the OG in 2D, $OG \perp AB$

[2]Users do have control over the relative position and relative orientation of surface pieces. The issue of global control will be discussed at the end of this chapter.

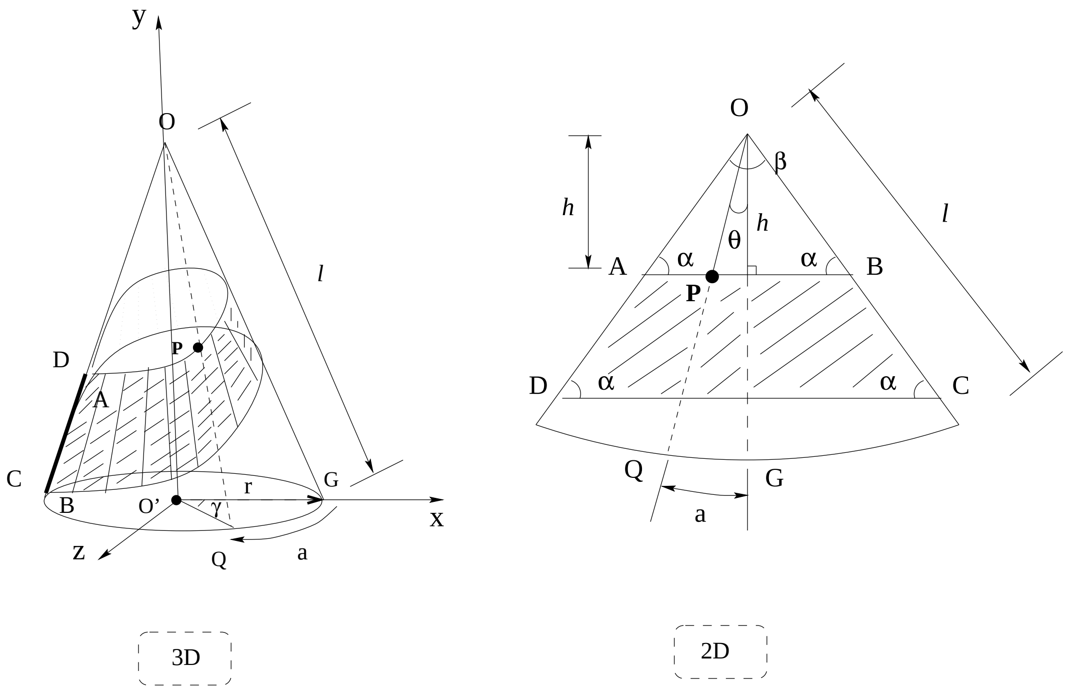


Figure 5.4: A simple developable surface.

in 2D. In the 2D configuration, h is the perpendicular distance from the apex O to the edge AB. As one can see in Figure 5.4,

$$|OP| = \frac{h}{\cos\theta},$$

$$a = l\theta.$$

Consider the angle $\gamma = \angle GO'Q$ in the 3D configuration,

$$\gamma = \frac{a}{r} = \frac{l\theta\ 2\pi}{l\beta} = \frac{2\pi\theta}{\beta},$$

where $\beta = \angle AOB = \pi - 2\alpha$ in 2D. Let (x_P, y_P, z_P) be the position of P and (x_Q, y_Q, z_Q) be the position of Q in 3D. Clearly,

$$\frac{x_P}{x_Q} = \frac{|OP|}{l} = \frac{h}{l\cos\theta}.$$

Hence,

$$\begin{aligned} x_P &= \frac{h}{l\cos\theta} x_Q \\ &= \frac{h}{l\cos\theta} r\cos\gamma \\ &= \frac{h}{l\cos\theta}\frac{l\beta}{2\pi}\cos\frac{2\pi\theta}{\beta} \\ &= \frac{h\beta}{2\pi\cos\theta}\cos\frac{2\pi\theta}{\beta} \end{aligned}$$

Similarly, we can show that

$$z_P = \frac{h\theta}{2\pi\cos\theta}\sin\frac{2\pi\theta}{\beta}.$$

Finally, we can determine y_P,

$$\frac{y_P}{y_Q} = -\frac{|OP|}{|OQ|}.$$

Therefore,

$$\begin{aligned} y_P &= \frac{-h}{l\cos\theta} y_Q \\ &= \frac{-h}{l\cos\theta} l\cos\alpha \\ &= \frac{-h}{\cos\theta}\frac{\sqrt{4\pi^2-\beta^2}}{2\pi} \end{aligned}$$

We can represent any point P on the developable surface by making h a parameter, where $h \in [h_{AB}, h_{DC}]$, h_{AB} is the distance from O to AB, and h_{DC} is the distance from O to DC in 2D. Now we have constructed a parametrization of the developable surface in terms of θ and h.

As we can see, this developable surface we considered above is a simple one. However, its analytic representation quickly becomes non-trivial. We can foresee how difficult it is to obtain the formulae for a more complex developable surfaces. It appears to be more prudent to obtain the discrete representation of the surface numerically.

5.4 Discussion

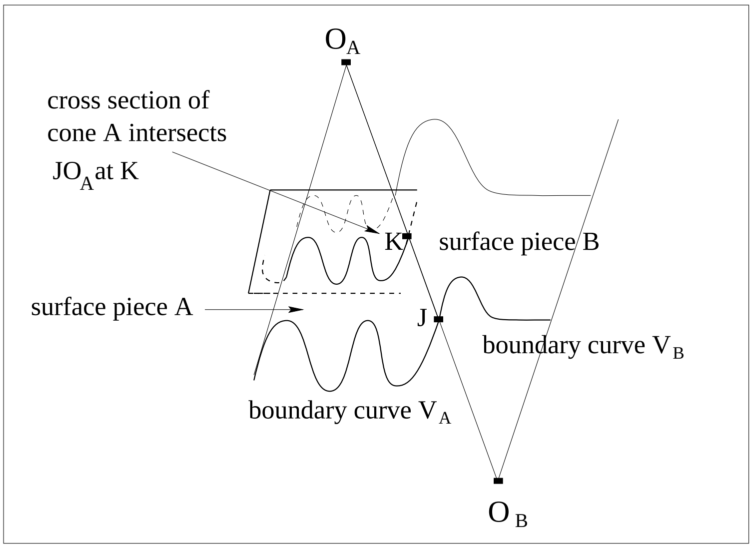
5.4.1 Local Shape Control

To change the shape of one surface piece, it is not necessary to subdivide the surface piece into smaller and smaller pieces. The user can adjust control points of the cross section curve of the cone associated to this surface piece. Since users can interactively change the piecewise continuous cross section curve of the generalized cone which defines the shape of associated surface piece, users have control over the local shape of each surface piece. Since developable surfaces are not stretchable, the adjustment of the shape of a surface piece might effect the relative orientation of this surface piece and its adjacent piece in the 3D space.

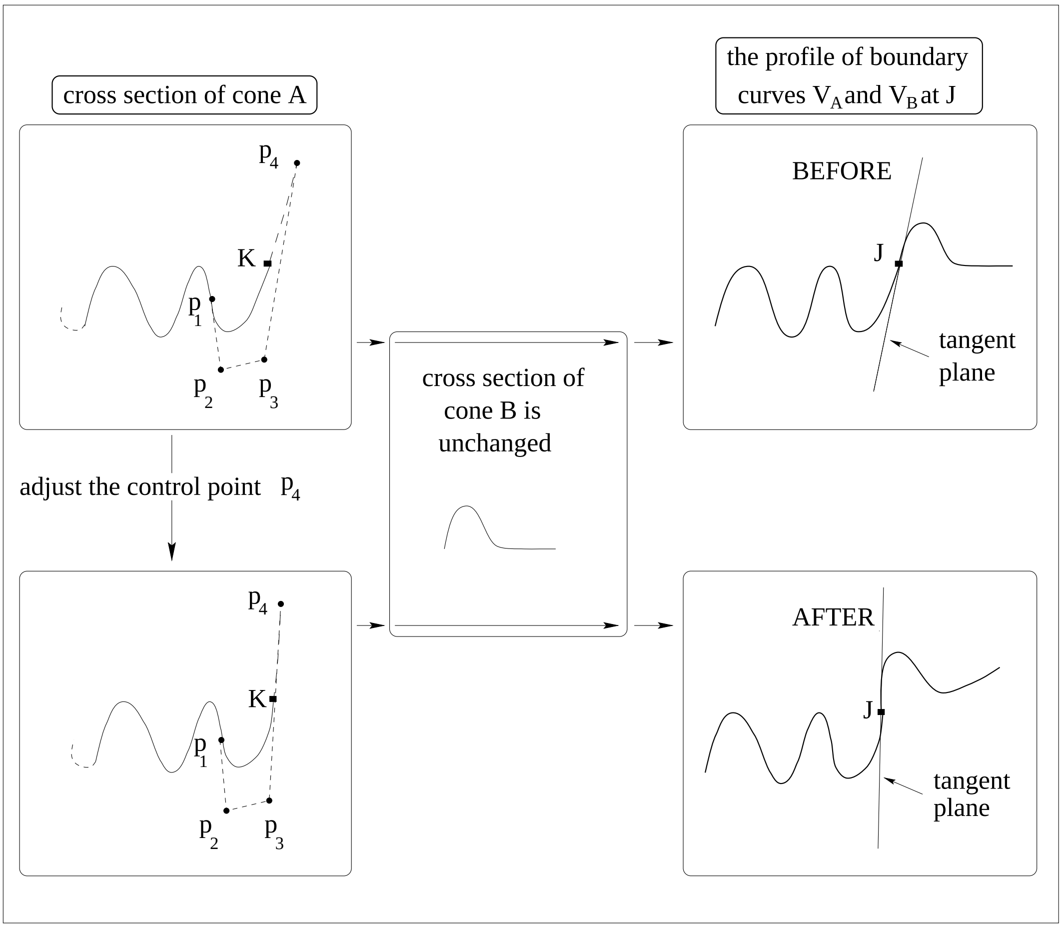
5.4.2 Global Control

The global control can also be achieved through adjustments of control points of cross section curves. For instance, we have two adjacent surface pieces A and B as shown in Figure 5.5(a). To control the relative orientation of surface pieces A and B, we can change the orientation of the tangent plane at their common generator. That can be done by adjusting a few Bézier control points of the user defined cross section curve of cone A, where cone A is associated to surface piece A. An example of such adjustment and its global effect is as shown in Figure 5.5(b). The boundary curve V_A of A and the boundary curve V_B of B join each other at point J. The cross section of cone A is as shown in the left column of Figure 5.5(b). Point K is the intersection point of generator JO_A and the cross section of cone A. Recall that the cross section curve is defined by piecewise cubic Bézier curves in our modelling system. Assume that point K is on the cubic Bézier defined by control points p_1, p_2, p_3 and p_4. When p_4 is adjusted, the tangent direction of the cross section curve at K might be changed as shown in the figure. As a consequence, the orientation of the tangent plane along the common generator O_AO_B is changed as shown in the right column of Figure 5.5(b). Therefore, the relative orientation of surface piece A and B are changed as one can tell from the profile of the boundary curves V_A and V_B at the neighbourhood of point J.

Notice that the local shape of each surface is not largely affected by this adjustment. Since the cross section curve of cone A is defined using piecewise continuous Bézier curve segments, moving control point p_4 affects the shape of only the last segment of the cross section curve. An user can choose to make this segment relatively small so that it can be used for global control, and he/she can still have local control over the shape of the resulting developable surface.



(a)



(b)

Figure 5.5: Adjusting one control point of a cross section curve to change the relative orientation of two surface piece A and B.

Chapter 6

Applications of the Model

In this chapter, we will demonstrate the feasibility of our approach. We will use our developable surface representation technique to model a hanging scarf and a looping structure.

6.1 Modelling a Hanging Scarf

6.1.1 Observed Properties of a Hanging Scarf

By a hanging scarf, we mean a scarf that is suspended in the air by one of its corners. If one lifts a silky scarf by a single corner, leaving the scarf hanging naturally in the air, such a hanging scarf is approximately a developable surface. Given a point on the scarf's surface, one can easily estimate roughly where the generator lies.

We can image an "invisible" generalized cone of a similar shape suspended in the air by its apex and the cross section of the cone as is shown in Figure 6.1(a). The corner of the scarf which the scarf is suspended by is close to the apex of the cone, and the scarf itself follows the contour of the cone.

The folds produced by the cross section, as shown in Figure 6.1(a), are expected to be very close to each other. In order to give the reader a better idea what the folds look like we will use the cross section in Figure 6.1(b) to demonstrate the design and modelling process in the following sections.

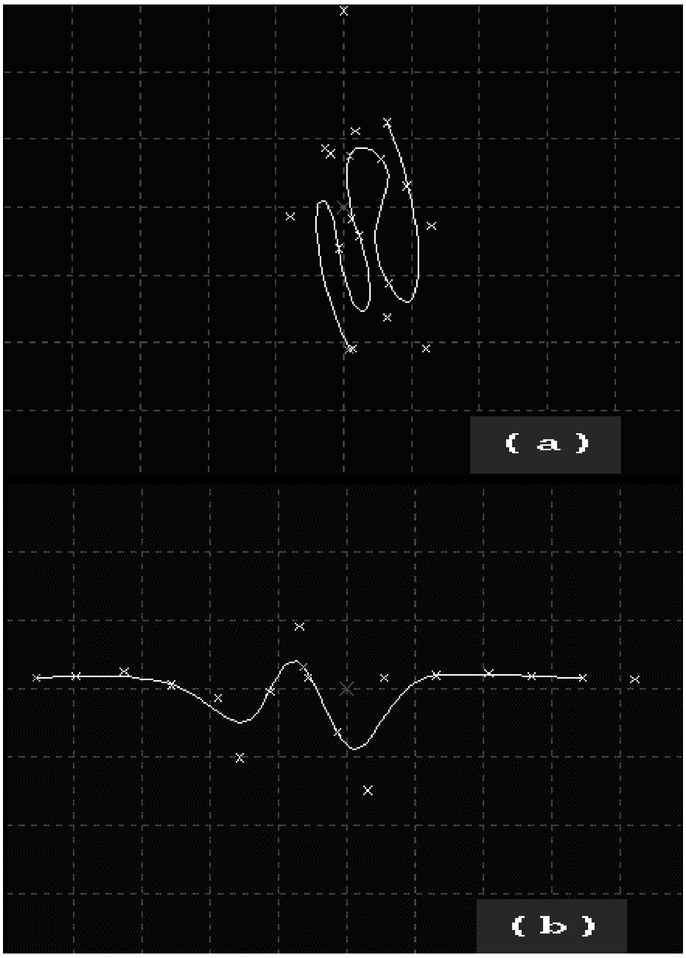


Figure 6.1: Two possible cross section designs (Hanging Scarf).

6.1.2 Approximating a Single Cross Section

By looking at a scarf, an intuition can be obtained for the folding of the hanging scarf. The point of departure for our system is an estimate what the horizontal cross section of the scarf looks like. The user designs the cross section in an interactive Cross Section Editor using cubic Bézier curves. The user may add or move Bézier control points. a neighbouring point might be adjusted automatically by the system, if necessary, to make sure that the resulting piecewise curve is C^1 continuous.

The approximated horizontal cross section of the scarf is used as the cross section of the "invisible" cone. The apex position of the cone in 3D is specified in a script file.

6.1.3 Defining Surface Pieces in 2D

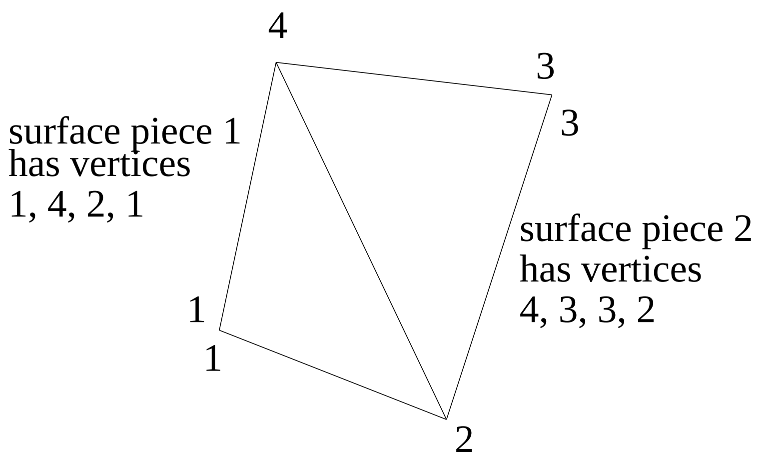


Figure 6.2: Surface piece partition (Hanging Scarf).

Since the developable surface can always be flattened out, we can define the scarf to-be-developed in 2D. The user defines the scarf by giving the coordinates of its vertices. Then the scarf is divided into two surface pieces, as shown in Figure 6.2.

Here, we will explain how the modelling system handles the surface piece in 2D. Without loss of generality, our system requires each surface piece to be defined by four vertices v_0, v_1, v_2 and v_3, where the line v_0v_3 is the first generator and the line v_1v_2 is the last generator of the surface piece. By the surface construction process described in chapter 5, the shape of each surface piece should be defined by one generalized cone. Surface pieces are treated differently depending on their shapes. Four general cases are considered:

CASE I: As shown in Figure 6.3, $v_0v_3||v_1v_2$ in the plane embedding the flattened surface. In this case, the shape of the developable surface in 3D should be defined by a generalized cylinder instead of a generalized cone. After development we should still have $v_0v_3||v_1v_2$ in 3D. We can show that this is true by way of contradiction.

Proof: Suppose that the shape of such a surface piece can be defined in 3D by a generalized cone. Since line v_0v_3 is the first generator and line v_1v_2 is the last generator of the surface piece, the two generators must intersect at the apex

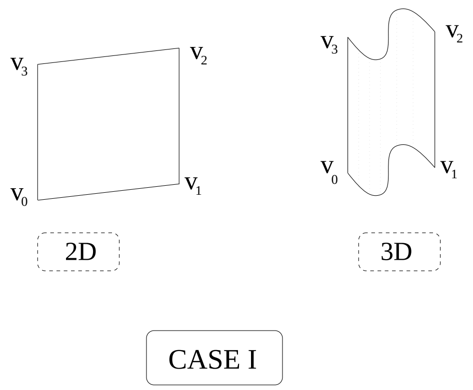


Figure 6.3: Surface piece type: Case I.

of the generalized cone. When the generalized cone and the surface piece are flattened out in a 2D plane, the one-to-one correspondence between points on the surface piece and points on the cone remains the same. Thus lines v_0v_3 and v_1v_2 intersect in the 2D plane. But this contradicts our assumption that $v_0v_3||v_1v_2$ in a 2D plane when the surface piece is flattened out. Therefore, the shape of the developable surface in 3D can not be defined by a generalized cone.

□

The shape of the developable surface in 3D can be defined by a generalized cylinder instead, in which case after development we have $v_0v_3||v_1v_2$ in 3D. In our system, this case is not implemented. To incorporate generalized cylinder into the model is easy, and it follows similar principles as those of generalized cones, i.e., flattening the cylinder out in 2D, finding the correspondence between points on a surface piece and points on the cylinder, and constructing the surface piece in 3D using this information.

CASE II: As shown in Figure 6.4, v_0v_3 and v_1v_2 intersect at a point p. Since the developed surface piece in 3D follows the contour of a cone, there should be a one-to-one correspondence between generators of the surface piece and generators of the associated cone patch. So point p should correspond to the apex of the cone in 2D after the cone is flattened out. The user can specify a special generator of the cone which corresponds to either v_0v_3 or v_1v_2. Then the

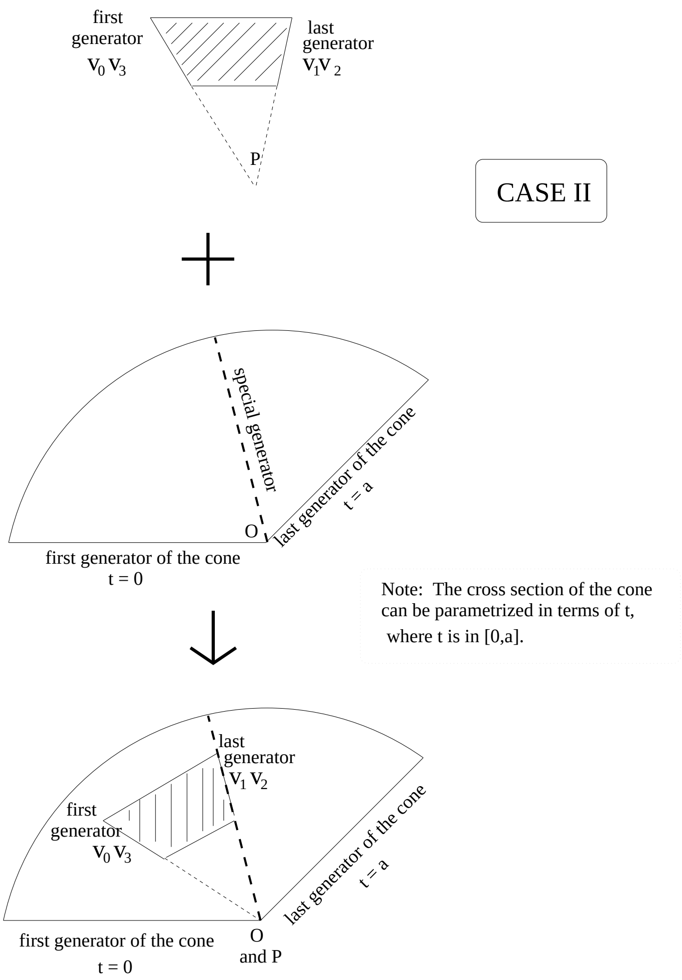


Figure 6.4: Surface piece type: Case II.

relation between the cone and the surface piece is determined as shown in the figure below. We will see this scenario in the looping structure example.

CASE III: As shown in Figure 6.5, v_0, v_1 and v_2 are distinct vertices, and v_3 coincides with v_0. Since v_1v_2 is a generator, the apex of the cone has to be on the line defined by v_1v_2. The user provides the distance from the apex to one of the vertices of the surface piece, in order to determine the relation between the surface piece and the cone.

CASE IV: The case where v_0, v_1 and v_3 are distinct vertices, and v_2 coincides with v_1 is symmetric to case III.

The situations described in case III and case IV occur in our hanging scarf example.

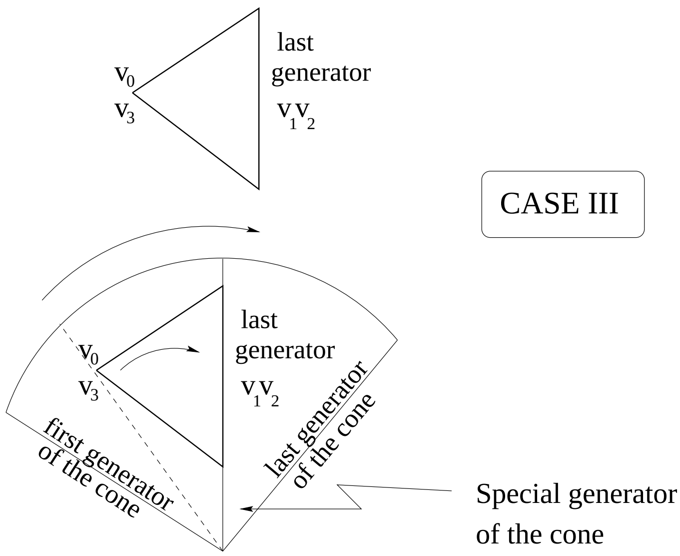


Figure 6.5: Surface piece type: Case III.

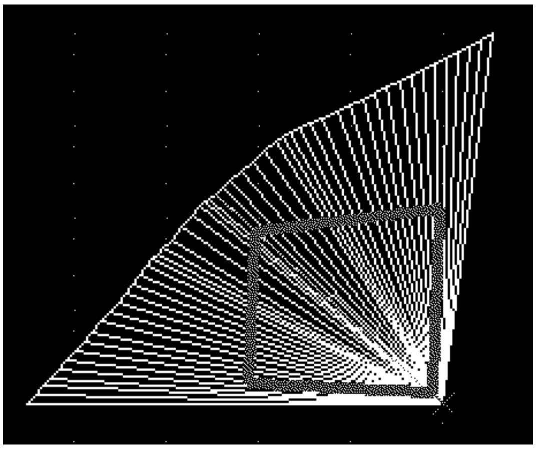


Figure 6.6: 2D mapping of the surface pieces and the cone (hanging scarf).

Figure 6.6 shows the 2D mapping of the two surface pieces and the cone, it also shows their relation in 2D. Note that the hanging scarf does not have to be a square. In this example run, an arbitrary quadrilateral is used.

6.1.4 Constructing Two Surface Pieces in 3D

We have discussed the method of constructing the developable surface in 3D in the previous chapter. The resulting surface is as shown in Figure 6.7. The configurations are: (a) the top view of the scarf, (b) the front view of it, (c) the view from an angle.

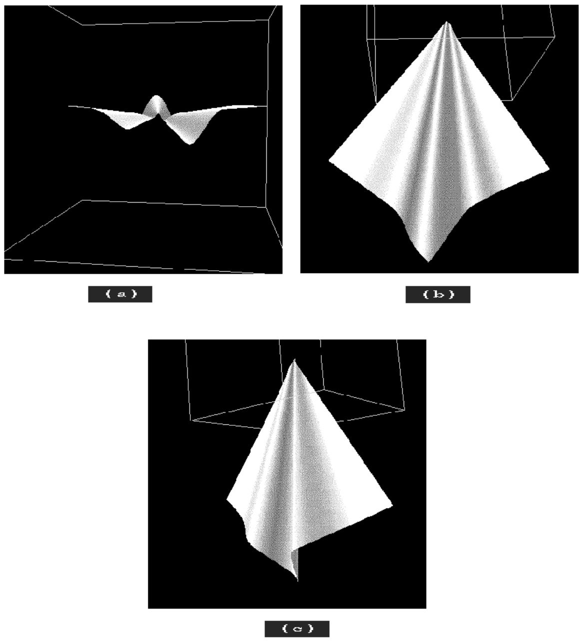


Figure 6.7: A hanging scarf: (a) top view; (b) front view; (c) view from an angle.

6.2 Modelling a Looping Structure

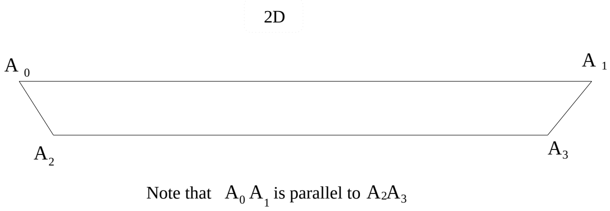


Figure 6.8: A long ribbon.

The looping structure example used in this section is a bow which resembles the type of bow used to decorate a gift box. We will "fold" a long ribbon into a bow, and the ribbon used here is as shown in Figure 6.8.

6.2.1 Observed Properties of a Looping Structure

There is a pattern in the structure of a bow. If we use an ellipse of roughly the same size to replace a loop in a bow, the pattern becomes more obvious. Some example patterns are shown in Figure 6.9. Therefore, it is sensible to define the shape of each

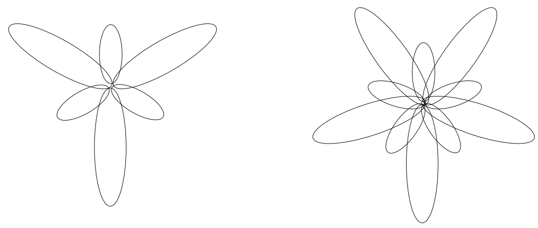


Figure 6.9: Example bow patterns.

loop using a generalized cone, and to introduce an intermediate piece to connect two adjacent loops if necessary.

6.2.2 Approximating the Cross Section

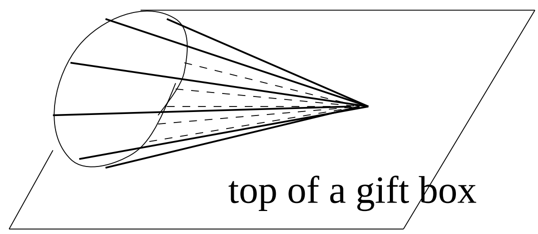


Figure 6.10: A cone used to define a loop.

To define a loop, one might use a cone which looks like the one shown in Figure 6.10. Here, we make the portion which is flattened as it touches the top of a gift box flatly, so that the resulting loop looks more realistic.

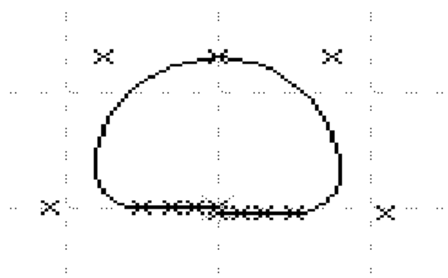


Figure 6.11: The **cross section** of the cone used for a loop.

We have constructed a cone cross section accordingly as shown in Figure 6.11. The intermediate pieces are triangular in 3D in this example, so the cone that defines their shape uses a straight line as its cross section.

6.2.3 Constructing the 2D Layout

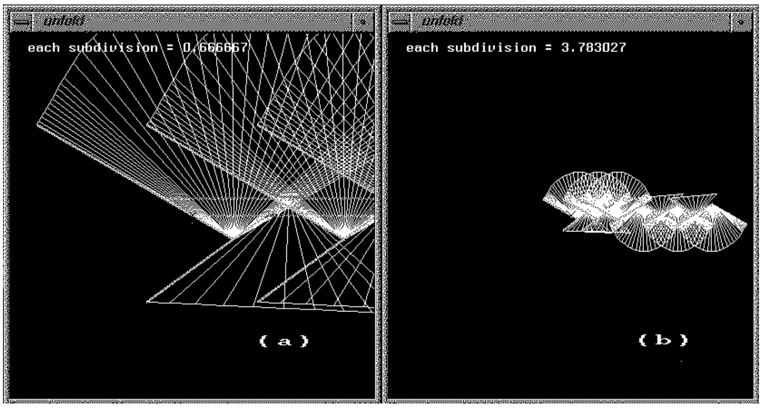


Figure 6.12: The 2D layout of the bow.

In this example, we will create three small loops and three big loops in our bow. We take a long ribbon, divide it into ten segments – six loop segments and four intermediate connecting segments. We draw the ribbon in the 2D layout window, rotate the 2D mapping of a cone to match its corresponding ribbon piece. Figure 6.12(b) is an overall picture of the ribbon pieces and the associated cones in 2D. Figure 6.12(a) shows some details of the 2D layout. The step by step construction of the 2D layout is as shown in Figure 6.13.

6.2.4 Constructing the 3D Layout

The method for constructing the developable surface in 3D is as described in the previous chapter. The surface pieces used in this example have a special form: we are able to approximate the surface using tensor-product Bézier patches. We will discuss this below. Note that this procedure may not be applied to surface pieces of arbitrary shapes.

In our model, we first approximate one particular edge of the ribbon in 3D by cubic Bézier curves. In chapter 3, we explained the advantage of using Bézier curves for design purposes. Higher degree Bézier curves might be used here, and we choose cubic Bézier curves for their simplicity. Then we use this information and take advantage of the special property of the ribbon piece to construct the desired Bézier patches.

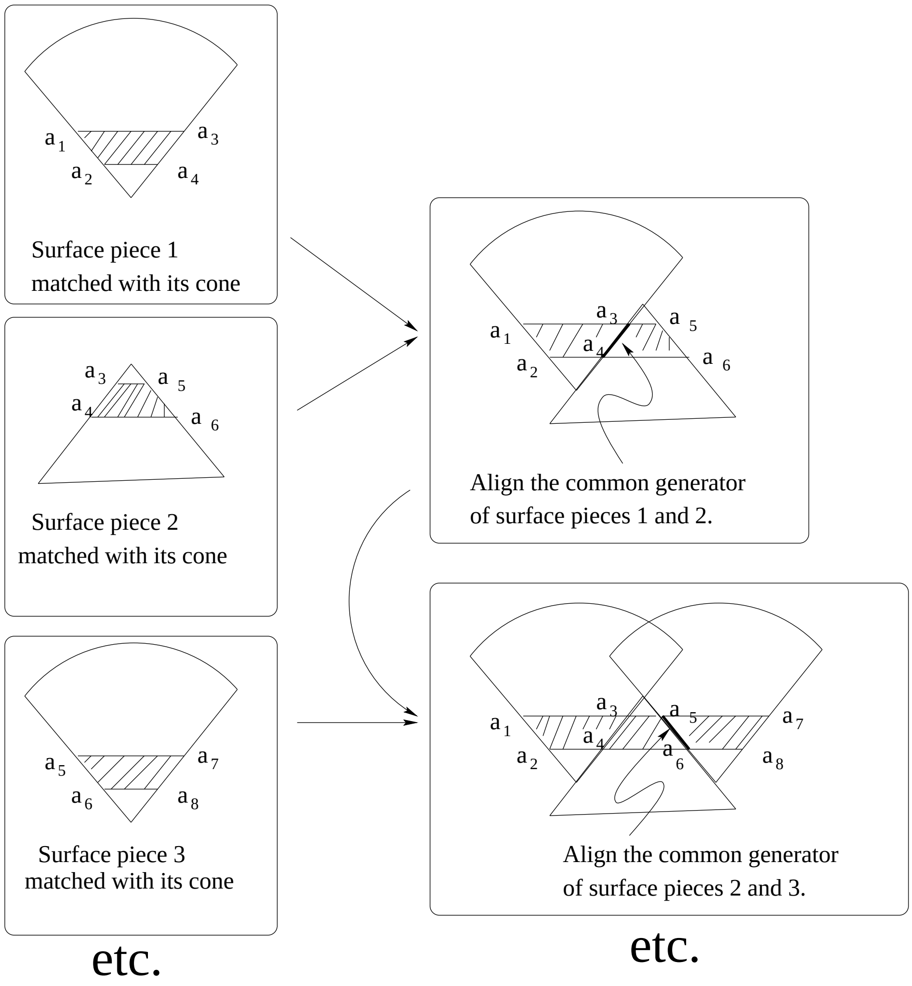


Figure 6.13: Step by step construction of the 2D layout of the looping structure.

First, we will show one way of adaptively approximating a given space curve using cubic Bézier curve segments. Then we will explain, knowing the cubic Bézier approximation of a segment of an edge, how a Bézier patch is created to approximate an original ribbon patch and why this procedure is valid for the example looping structure.

6.2.4.1 Approximating a Space Curve by G^1 Continuous Cubic Bézier Curve Segments

Given a space curve $Q(t)$, $t \in [0,1]$, and an error limit ERR, we would like to construct a piecewise continuous cubic Bézier curve which approximates the original curve to within some error tolerance ERR.

Procedure *Approximate_curve* (Q, a, b, ϵ)
/* $Q(t)$ is the original curve, where $t \in [0, 1]$.
/* We consider the segment of Q from $t = a$ to $t = b$.
/* The procedure returns a piecewise continuous cubic Bezier curve
/* which approximate this segment of Q and the sum of square error
/* is bounded by the ϵ.

T_a = the tangent of the curve at $Q(t = a)$.
T_b = the tangent of the curve at $Q(t = b)$.
Let c be an unknown constant and $c > 0$.
Construct a cubic Bezier curve B with the control points
$Q(a)$, $Q(a) + cT_a$, $Q(b) - cT_b$, $Q(b)$.
Compare B and the original curve segment, and
express the least of square error in terms of c, which gives
us a quadratic function in c.
Minimize this over *err_min*, by finding
the minimal of the quadratic function, provided that $c > 0$.

If (*err_min* $< \epsilon$) then
we are done, return the cubic Bezier curve B.
else
Approximate_curve (Q, a, $\frac{a+b}{2}$, $\epsilon/2$)
Approximate_curve (Q, $\frac{a+b}{2}$, b, $\epsilon/2$)
end_if
end_Approximate_curve

Figure 6.14: Pseudocode for **procedure** *Approximate_curve*.

In our model, this is done using the recursive procedure shown in Figure 6.14. Intuitively, if we can find a satisfactory cubic Bézier approximation of the given curve segment, this cubic Bézier representation is returned. Otherwise, the given curve segment is split into two halves and each half is approximated using the same procedure. This recursive procedure (**Procedure** *Approximate_curve*) is shown in Figure 6.14.

To obtain the piecewise cubic Bézier approximation to the curve Q, we simply call

$$Approximate\_curve\ (Q,\ 0,\ 1,\ ERR).$$

Recall that during the bow construction process, we divided the ribbon into a num-

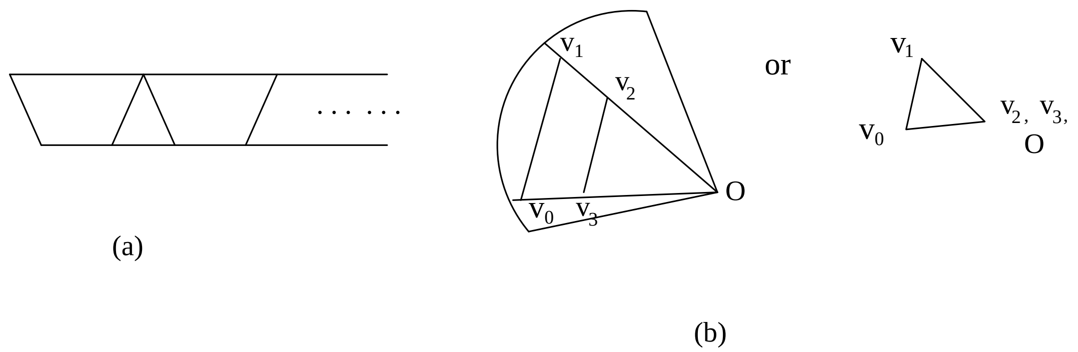


Figure 6.15: Ribbon surface pieces.

ber of pieces as shown in Figure 6.15(a). Now, for each ribbon piece, we choose one boundary curve, use the above procedure to find a reasonable cubic Bézier approximation to it. In the next subsection, we will explain how this edge is chosen and how its Bézier approximation can be used to construct the desired Bézier patch.

6.2.4.2 Construct Tensor-product Bézier Patches

In this section, tensor-product Bézier patches will be constructed to approximate the developable ribbon surface. Our ribbon pieces have some special properties. As shown in Figure 6.15(a), the top and the bottom edges of the ribbon are parallel. Assume that we name the corner of each piece $v_0v_1v_2v_3$ as shown in Figure 6.15(b), where O is the apex of the associated cone, $|v_3O| < |v_0O|$, $|v_2O| < |v_1O|$, v_0v_3 and v_1v_2 are generators of the ribbon surface. Here, we have either $v_0v_1||v_3v_2$ or v_2 coincides with v_3.

As one can see from Figure 6.15(b), if the triangle Ov_0v_1 is scaled by a factor, namely

$$k = \frac{|Ov_3|}{|Ov_0|},$$

the resulting triangle is congruent to the triangle Ov_3v_2. Similarly, in 3D, if the curved ribbon piece boundary v_0v_1 is scaled by k, then the resulting curve is congruent to the piece boundary v_3v_2.

A similar relationship can be established between tangents of corresponding points

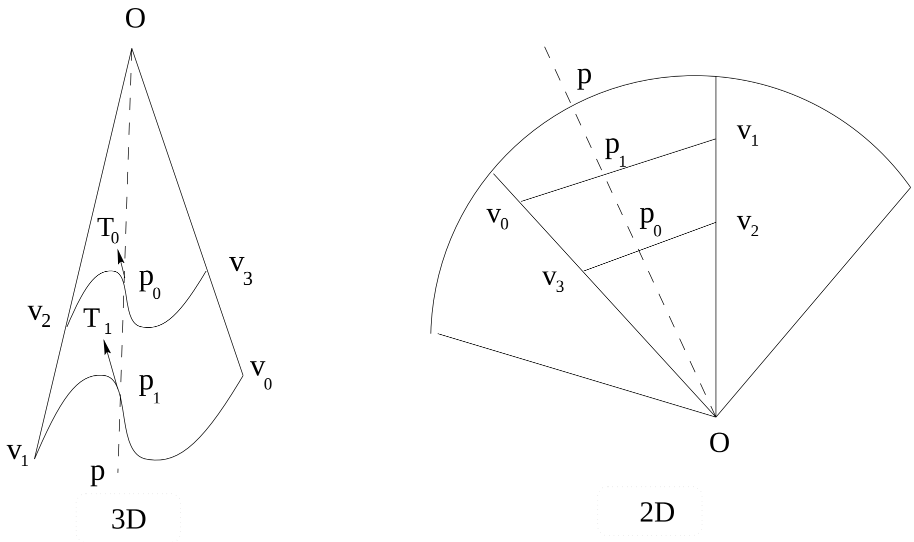


Figure 6.16: Corresponding points p_0 and p_1.

of boundary curves v_0v_1 and v_3v_2. Pick a generator of the cone which intersects the ribbon piece boundary v_3v_2 at p_0 and v_0v_1 at p_1, as shown in Figure 6.16. Let T_0 denote the tangent vector of the boundary v_3v_2 at p_0 and T_1 denote the tangent vector of the boundary v_0v_1 at p_1. If O is at the origin and the boundary curve v_0v_1 in 3D is scaled by a factor, k, then the resulting curve matches the boundary curve v_3v_2 exactly. In other words, if the parametric form of curve v_0v_1 is $\mathbf{B}_1(t)$, $t \in [a, b]$, then curve v_3v_2 can be parametrized by $k\mathbf{B_1}(t)$, $k \geq 0$. If point

$$p_1 = \mathbf{B_1}(t^*),$$

for an some $t^* \in [a, b]$, then

$$p_0 = k\mathbf{B_1}(t^*), \quad T_1 = \dot{\mathbf{B}}_1(t^*), \quad T_0 = k\dot{\mathbf{B}}_1(t^*).$$

Therefore,

$$\frac{|T_0|}{|T_1|} = k.$$

We have

$$\frac{|T_0|}{|T_1|} = \frac{|Op_0|}{|Op_1|} = \frac{|Ov_3|}{|Ov_0|} = k. \tag{6.1}$$

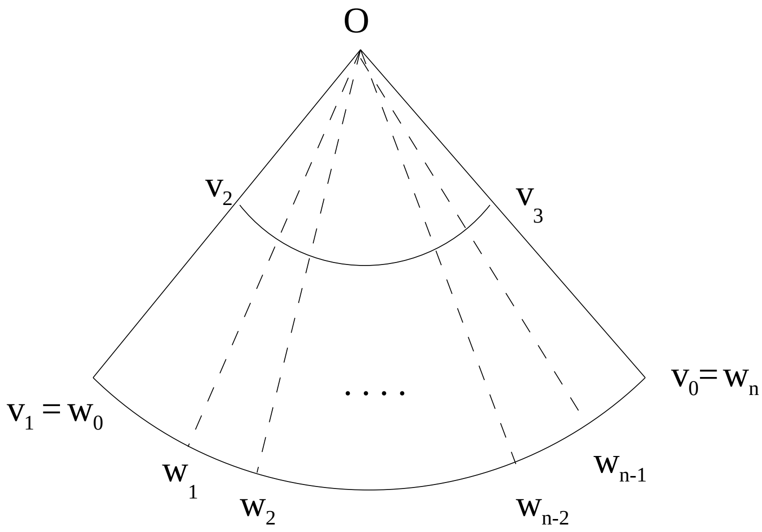


Figure 6.17: Sub-dividing a ribbon piece into n ribbon patches.

Now, consider the ribbon piece $v_0v_1v_2v_3$ shown in Figure 6.17. We pick the boundary v_0v_1 and approximate it with n cubic Bézier curves using **procedure** *Approximate_curve*. Note that although the cross section of the corresponding cone is a piecewise continuous Bézier curve, the boundary of a ribbon piece is not defined as a Bézier curve. So this boundary curve approximation step is necessary. The curves' endpoints are

$$v_1 = w_0, w_1, ..., w_{n-1}, w_n = v_0,$$

where w_{i-1} and w_i are the endpoints of the i^{th} Bézier curve, $i = 1, 2, ..., n$. Lines Ow_i, $i = 1, 2, ..., n$, divide the ribbon piece into n *ribbon patches*.

Consider a single ribbon patch $q_0q_1r_1r_0$, as shown in Figure 6.18.[1] Let B_1 denote the cubic Bézier representation of the boundary curve q_1r_1. Let q_1, g_1, h_1 and r_1 denote the control points of B_1. We approximate the boundary curve q_0r_0 by the cubic Bézier curve B_0 defined by control points q_0, g_0, h_0 and r_0, where

$$q_0g_0 = k\ q_1g_1 \quad and \quad h_0r_0 = k\ h_1r_1.$$

By the algorithm in **procedure** *Approximate_curve*, the vector r_1h_1 is parallel to the

[1]It is possible for the points q_0 and r_0 to coincide.

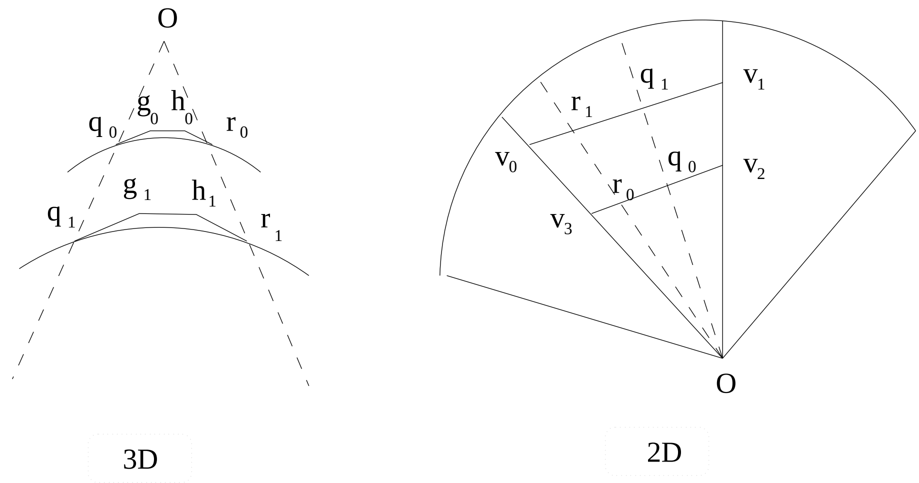


Figure 6.18: One ribbon patch.

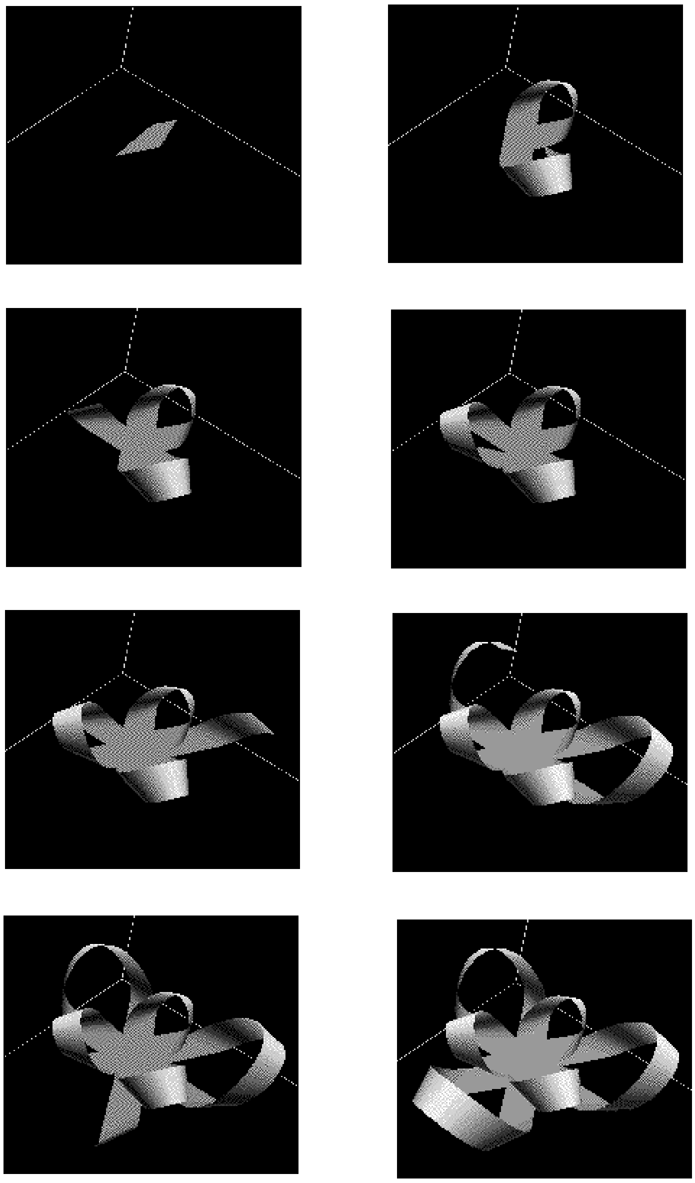
tangent of the boundary r_1q_1 at r_1. Therefore, by equation 6.1, the vector r_0h_0 is parallel to the tangent of the boundary r_0q_0 at r_0. Similarly, the vector g_0q_0 is parallel to the tangent of the boundary r_0q_0 at q_0.

Assume the cubic Bézier curves are both parametrized in terms of t. There is a one-to-one correspondence between the point $B_0(t)$ and the point $B_1(t)$. The line defined by $B_0(t)$ and $B_1(t)$ is an approximate generator of the surface. So we can define a tensor-product Bézier patch of degree $(3, 1)$ using the following set of control points

$$\begin{array}{cccc} q_0 & g_0 & h_0 & r_0 \\ q_1 & g_1 & h_1 & r_1 \end{array}.$$

We approximate each ribbon patch by a Bézier patch to obtain the Bézier representation of the bow in 3D.

6.2.5 Resulting Surface



The development of the bow in 3D.

The figure above shows the development of a surface in 3D. The ribbon consists of 10 surface pieces. When the sum of square error limit is set to 0.0005, each surface piece is approximated by 1 or 2 or 4 ribbon patches. For our loop structure, the flatter a surface piece is, the fewer ribbon patches are needed for the approximation.

6.3 Discussion

The two examples presented in this chapter have demonstrated the feasibility of the technique. The design of the modelling tool can be flexible. For instance, the current version of the modelling tool allows an user to interactively define the cross section of a cone associated with a surface piece. We can add an option which allows the user to define the cone using a space curve $\alpha(t)$ and an apex point O, because the family of lines $O\alpha(t)$ spans a generalized cone as well. This feature might be useful for some surface construction tasks. In the next chapter, we will discuss possible extensions of the modelling tool.

Chapter 7

Conclusion

7.1 Contributions of the Thesis

The main contribution of this thesis is to introduce a new approach to the design and modelling of a class of developable surfaces. By dividing the surface into a number of pieces and modelling each piece with a generalized cone, the user has a great deal of control over the shape of the developable surface. Unlike the previous approaches described in chapter 4, the user can specify the cone by the shape of its cross section and the position of its apex, so it is easy to determine a set of input data which lead to a good approximation of the desired surface. The concept is simple and easy to grasp, but it does not reduce the power of our modelling system. The user can interactively change the input to the system. The feasibility of our model is demonstrated by applying to it to the modelling of a hanging scarf and a bow, as shown in Figures 6.7 and 6.2.5.

To obtain a better approximation of the desired developable surface, it is not usually necessary to subdivide each surface piece into smaller and smaller surface patches. To change the shape of one surface piece, the user can adjust control points of the cross section curve of the cone associated to this surface piece. Also, by editing a few controls points, the user can also alter the orientation of the tangent plane at a shared boundary of two adjacent surface pieces. Therefore, this approach gives the user a great deal of control over the shape of the resulting developable surface, both locally and globally. This is discussed at the end of chapter 5.

7.2 Extensions to the System

Our modelling system can be extended to handle developable surfaces whose surface pieces can be defined using generalized cylinders as well as generalized cones. By improving the user interface and the overall design of our system, our modelling tool should be able to handle a variety of developable surfaces. For instance, if a sheet is not polygonal, i.e., its boundaries can not be defined by straight line segments, we might need to use curves to describe its boundaries. Another extension to the system is to approximate applicable surfaces, such as crumpled paper or paper with sharp folds and cusps, by gluing a number of developable surface pieces together. Also, we might allow the user to specify a space curve, and let the edge of a developable surface follow the space curve as closely as possible without tearing or wrinkle. Such a feature might enable us to create many interesting developable surface animations. An important extension would be collision detection can be added to the system to prevent surface inter-penetration. Volino, Courchesne and Thalmann [13] presented interesting collision detection techniques for cloth, fabrics or, more generally, deformable surfaces. This technique might be useful in an extension of our developable surface modelling system.

7.3 Areas of Further Research

An interesting area of further research might be to incorporate physical constraints into the model. For instance, we might want to simulate the movement of a scarf. Imagine that the scarf is suspended in the air by its corner C and we hold the corners of the scarf adjacent to C, so that all four edges are straight in the air as shown in Figure 7.1. Then we release corners adjacent to C, and let them fall. To simulate this, we could use physical constraints to calculate the force and surface energy, so that the system can generate the behaviour of the scarf automatically. Another example is

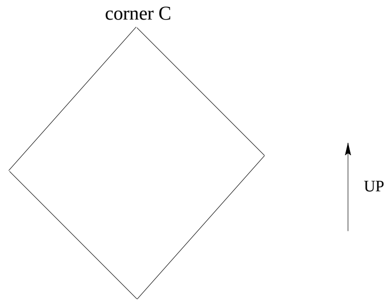


Figure 7.1: An example initial configuration of a scarf in the air.

that of placing a bow on a table and pressing down on it. If we remove the pressure, the loops of the bow should spring back and relax to its original configuration. Notice that the behaviour of the developable surface under physical constraints are dependent on the material. Also, the realistic simulation of non-rigid objects such as the ones mentioned here needs to be formalized in a rigorous mathematical framework. By taking the physical constraints into consideration, our modelling system would be more interesting and complete.

Bibliography

[1] Günter Aumann, "Interpolation with Developable Bézier Patches", *Computer Aided Geometric Design*, Vol. 8, (1991), pp 409-420.

[2] M. Do Carmo, "Differential Geometry of Curves and Surfaces", Prentice Hall, Englewood Cliffs, NJ, U.S.A., 1976.

[3] Gerald Farin, *Curves and Surfaces for Computer Aided Geometric Design — A Practical Guide*, Third Edition, Academic Press, 1992.

[4] I. D. Faux and M. J. Pratt, "Computational Geometry for Design and Manufacture", Ellis Horwood, Chichester, UK, 1980.

[5] E. Fiume, "Isometric Piecewise Polynomial Curves", *Computer Graphics Forum*, Vol. 14, (January, 1995), pp 47-58.

[6] James D. Foley, Andries van Dam, Steven K. Feiner and John F. Hughes, *Computer Graphics – Principles and Practice*, Second Edition, Addison-Wesley Publishing Company, Reading, Massachusetts, U.S.A., 1990.

[7] D. Hilbert and S. Cohn-Vossen, *Geometry and the Imagination*, Chelsea, New York, NY, U.S.A., 1952.

[8] Josef Hoschek and Dieter Lasser, translated by Larry L. Schumaker, *Computer Aided Geometric Design*, A K Peters, Wellesley, Massachusetts, 1993.

[9] Yannick L. Kergosien, Hironobu Gotoda and Tosiyasu L. Kunii, "Bending and Creasing Virtual Paper", *IEEE Computer Graphics and Applications* (January 94), pp 40-48.

[10] A. W. Nutbourne, P. M. McLellan and R. M. L. Kensit, "Curvature Profiles for Plane Curves", *Computer-Aided Design*, Vol. 4, No. 4, (July, 1992), pp 176-184.

[11] P. Redont, "Representation and Deformation of Developable Surfaces", *Computer Aided Design*, Vol. 21, No. 1, (Jan/Feb, 1989), pp 13-20.

[12] John A. Roulier, "Specifying the Arc Length of Bézier Curves", *Computer Aided Geometric Design*, Vol. 10, (January, 1993), pp 25-56.

[13] Pascal Volino, Martin Courchesne and Nadia Magnenat Thalmann, "Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects", *Proceedings of SIGGRAPH '95*, (August, 1995), pp 137-144.

[14] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques*, Addison-Wesley, New York, N.Y., 1992.

[15] Gunter Weiss and Peter Furtner, "Computer-aided Treatment of Developable Surfaces", *Computer & Graphics*, Vol. 12, No. 1, (1988), pp 39-51.