Dual Space Drawing Methods for Ruled Surfaces with Particular Shapes

Rena Ding and Yan Zhang
email: {yding,yan}@cit.uws.edu.au
University of Western Sydney, Australia

Summary

Ruled surface design and its applications are widely used in CAGD (Computer Aided Geometric Design) and industrial areas. The hyperbolic paraboloid, the cylinder, the cone, and the hyperboloid are ruled surfaces with particular shapes. In this paper, dual space drawing methods for these particular ruled surfaces are proposed. The underlying principles of these methods are the applications, extensions and variations of the dual (space) de Casteljau algorithm [15].The hyperbolic paraboloid is created by using the initial dual de Casteljau algorithm directly. The cylinder, the cone, and the hyperboloid, on the other hand, are created by using the extensions or variations of the dual de Casteljau algorithm. During the drawing processes, the screen representation of the rulings is presented. This treatment presents the rulings as clipped line segments on the computer screen in order to show the proportion of the ruled surface on the computer screen. In addition, the dual characteristics of Bezier surfaces are also discussed. The special construction method of the displacement matrix with two parallel control screws is demonstrated. The normal screw is normalized into the unit screw as well. These new methods show the universal possibility of the dual space approach for drawing ruled surfaces.[1]

*Key Words***:**
dual space, de Casteljau algorithm, ruled surface, Bezier surface

1 Introduction

Recently, the dual space design methods of curves and surfaces have been extensively studied. B-splines in dual quaternion spaces are applied in [16]. The authors used rational B-splines in the space of quaternions to deal with curve approximation problems and path-smoothing, speed-smoothing problems. Juttler [7] used quaternion representation of RMM (Rotation Minimizing Motions) to sweep surface modeling. Ding [3] applied the De Boor algorithm in 6-dimensional space to draw particular ruled surfaces, namely, duplicated surfaces. Leopoldseder [8] used dual methods to draw cones. Pottmann and Wallner [13] used a dual approach to solve the drawing methods of differential geometry of developable surfaces, which include ruled surfaces. Our work is inspired by the paper of Sprott and Ravani [15], in which the de Casteljau algorithm in screw spaces was introduced to resolve ruled surface drawing and mesh generation problems. In this paper, we use the dual de Casteljau algorithm to improve and further develop their methods to draw particular ruled surfaces.

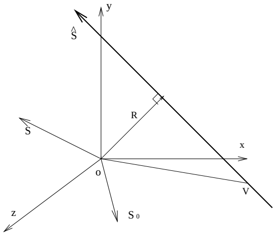


Figure 1. Geometrical interpretation of screw $\hat{\mathbf{S}} = (1 + \varepsilon p)\mathbf{S} + \varepsilon\mathbf{S_0}$ in 3-D.

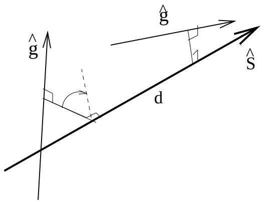


Figure 2. Displacement of line $\hat{\mathbf{g}}$ by translation and rotation along axis $\hat{\mathbf{S}}$ by $\hat{\theta} = \theta + \epsilon d$.

2 Screw Theory

2.1 Definition of a Screw

A general *screw* $\hat{\mathbf{S}}$, from which we derive a *line* by specialisation, consists of two parts, a real 3-vector $\mathbf{S}$ which indicates the *direction* of (line of) the screw, and a real 3-vector $\mathbf{S_p}$ which locates $\hat{\mathbf{S}}$ by recording the *moment* of the screw about the origin [9, 10]. In these terms, a screw is formed as

$$\hat{\mathbf{S}} = \mathbf{S} + \varepsilon\mathbf{S_p} = \mathbf{S} + \varepsilon(p\,\mathbf{S} + \mathbf{S_0}), \qquad (1)$$

where $\mathbf{S_0} = \mathbf{R} \times \mathbf{S} = \mathbf{V} \times \mathbf{S}$, and in which ε is a quasi-scalar satisfying $\varepsilon^2 = 0$. Here, p is the *pitch* of the screw and $\mathbf{S_0}$, which is the moment of the *line of the screw* about the origin, is orthogonal to $\mathbf{S}$ (*i.e.*,$\mathbf{S} \cdot \mathbf{S_0} = 0$), and is derived as shown from the origin radius vector $\mathbf{R}$ or, more generally, from any point $\mathbf{V}$ of the screw. The length $|\mathbf{S}|$ of the real part $\mathbf{S}$ is the real *magnitude* of the screw. Figure 1 illustrates a geometric interpretation of a screw.

2.2 Zero Pitch and Normalization of a Screw

Since a screw is a line with an associated pitch value p, it follows that a *line*, in its own right, is a screw with zero-valued pitch, *i.e.*, $p = 0$. Given a general screw $\hat{\mathbf{S}}$, we often need access to its *normalized* line of unit magnitude and zero pitch. Derivation of such a *unit line*, which we denote specifically by the lower case form $\hat{\mathbf{s}} = \mathbf{s} + \varepsilon\mathbf{s_0}$, is straightforward. By equation (1),

$$\hat{\mathbf{S}} = (1 + \varepsilon p)\mathbf{S} + \varepsilon\mathbf{S_0}, \quad where \quad \mathbf{S_0} \cdot \mathbf{S} = 0.$$

Since $\varepsilon^2 = 0$, we find that

$$\hat{\mathbf{S}} \cdot \hat{\mathbf{S}} = (1 + \varepsilon p)^2\mathbf{S} \cdot \mathbf{S}, \quad so \quad \sqrt{\hat{\mathbf{S}} \cdot \hat{\mathbf{S}}} = (1 + \varepsilon p)|\mathbf{S}|.$$

[1] Manuscript received January,2006

Thus, the general screw $\hat{\mathbf{S}}$ normalises to its unit line $\hat{\mathbf{s}}$ in the form

$$\hat{\mathbf{s}} = \frac{\hat{\mathbf{S}}}{\sqrt{\hat{\mathbf{S}} \cdot \hat{\mathbf{S}}}} = \frac{1-\varepsilon p}{|\mathbf{S}|}\hat{\mathbf{S}} = \frac{1}{|\mathbf{S}|}(1-\varepsilon p)[(1+\varepsilon p)\mathbf{S} + \varepsilon \mathbf{S_0}] \\ = \frac{\mathbf{S}}{|\mathbf{S}|} + \varepsilon \frac{\mathbf{S_0}}{|\mathbf{S}|}. \qquad (2)$$

When, in a practical application, we are given a screw of the form $\hat{\mathbf{S}} = \mathbf{S} + \varepsilon \mathbf{S_p}$, we may derive its *pitch*, *origin radius*, and *normalised line* as

$$p = \frac{\mathbf{S} \cdot \mathbf{S_p}}{|\mathbf{S}|^2}, \qquad \mathbf{R} = \frac{\mathbf{S} \times \mathbf{S_p}}{|\mathbf{S}|^2}, \qquad \hat{\mathbf{s}} = \frac{\mathbf{S}}{|\mathbf{S}|} + \varepsilon \frac{\mathbf{S_p} - p\mathbf{S}}{|\mathbf{S}|} \qquad (3)$$

respectively, the last of these, equivalent to equation (2), making use of the first. We observe that a screw $\hat{\mathbf{S}}$ for which $\mathbf{S} = \mathbf{0}$ is a *line at infinity*. In this paper, all general screws are normalized to their unit lines before displacement, as described in [3].

2.3 A Screw or Line is a dual 3-Vector

When the component 3-vectors of $\hat{\mathbf{S}}$ are made explicitly, as $\mathbf{S} = (L_x, L_y, L_z)$ and $\mathbf{S_p} = (M_x, M_y, M_z)$, we find that a screw is not simply a 6-tuple of real Plűcker co-ordinates $L_x, L_y, L_z, M_x, M_y, M_z$. It is also a 3-vector

$$\hat{\mathbf{S}} = \mathbf{S} + \varepsilon \mathbf{S_p} = (L_x + \varepsilon M_x, L_y + \varepsilon M_y, L_z + \varepsilon M_z) \qquad (4)$$

in which each element, such as $L_x + \varepsilon M_x$, *etc.* is a *dual number*.

In these *dualised* terms, 3-vector quantities which are familiar in real co-ordinates acquire spatial interpretations. The *scalar product* $\hat{\mathbf{S}}_1 \cdot \hat{\mathbf{S}}_2$ of such 3-vectors comprises the cosine of the *dual angle* (see equation (6)), including both real angle and distance, between the screws $\hat{\mathbf{S}}_1$ and $\hat{\mathbf{S}}_2$. When $\hat{\mathbf{S}}_1$ and $\hat{\mathbf{S}}_2$ are *orthogonal*, so that $\hat{\mathbf{S}}_1 \cdot \hat{\mathbf{S}}_2 = 0 + \varepsilon 0 = 0$, these screws intersect one another at right angles. Similarly, the *vector product* $\hat{\mathbf{S}}_1 \times \hat{\mathbf{S}}_2$ is a screw which lies on the *common perpendicular* of the screws $\hat{\mathbf{S}}_1$ and $\hat{\mathbf{S}}_2$.

Most importantly, screws transform as 3-vectors do [10]. If $\hat{\mathbf{x}}_i$, $\hat{\mathbf{y}}_i$ and $\hat{\mathbf{z}}_i$ are mutually orthogonal unit lines defining a reference frame, *i.e.*, $\hat{\mathbf{x}}_i^2 = \hat{\mathbf{y}}_i^2 = \hat{\mathbf{z}}_i^2 = 1, \quad \hat{\mathbf{x}}_i \cdot \hat{\mathbf{y}}_i = \hat{\mathbf{y}}_i \cdot \hat{\mathbf{z}}_i = \hat{\mathbf{z}}_i \cdot \hat{\mathbf{x}}_i = 0, \quad \hat{\mathbf{x}}_i \times \hat{\mathbf{y}}_i = \hat{\mathbf{z}}_i$, the coordinates of a general screw $\hat{\mathbf{S}}$, when expressed in that i-frame, are

$$\hat{\mathbf{S}}_i = \begin{bmatrix} \hat{\mathbf{x}}_i^T \\ \hat{\mathbf{y}}_i^T \\ \hat{\mathbf{z}}_i^T \end{bmatrix} \hat{\mathbf{S}} = \begin{bmatrix} \hat{\mathbf{x}}_i \cdot \hat{\mathbf{S}} \\ \hat{\mathbf{y}}_i \cdot \hat{\mathbf{S}} \\ \hat{\mathbf{z}}_i \cdot \hat{\mathbf{S}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_i \cdot \mathbf{S} \\ \mathbf{y}_i \cdot \mathbf{S} \\ \mathbf{z}_i \cdot \mathbf{S} \end{bmatrix} + \varepsilon \begin{bmatrix} \hat{\mathbf{x}}_i \otimes \hat{\mathbf{S}} \\ \hat{\mathbf{y}}_i \otimes \hat{\mathbf{S}} \\ \hat{\mathbf{z}}_i \otimes \hat{\mathbf{S}} \end{bmatrix}, \qquad (5)$$

in which, for example, $\hat{\mathbf{x}}_i \otimes \hat{\mathbf{S}} = \mathbf{x}_i \cdot \mathbf{S_p} + \mathbf{x_0}_i \cdot \mathbf{S}$, where $\hat{\mathbf{x}}_i = \mathbf{x}_i + \varepsilon \mathbf{x_0}_i$ and $\mathbf{x}_i \cdot \mathbf{x_0}_i = 0$.

2.4 The Displacement of a Screw

2.4.1 The General Displacement of a Screw

Under the dualisation just described, points of linear interpolation between two given real points find spatial analogs in the locations to which a screw successively moves when displaced from some initial to some final location about a *screw axis* $\hat{\mathbf{s}}$ – see Figure 2. Such displacements are parameterised by a *dual angle* $\hat{\theta} \equiv \theta + \epsilon d$ which comprises the real angle θ of rotation, and the real distance d of translation, as measured about the axis $\hat{\mathbf{s}}$. For such a dual angle we may write

$$\sin\hat{\theta} \equiv \sin\theta + \epsilon d \cos\theta, \qquad \cos\hat{\theta} \equiv \cos\theta - \epsilon d \sin\theta. \qquad (6)$$

By dualising equations for purely rotational displacement [9], we find that a screw $\hat{\mathbf{X}}$ displaced from an initial location $\hat{\mathbf{X}}_0$ through dual angle $\hat{\theta}$ about a screw axis $\hat{\mathbf{s}} = (\hat{s}_x, \hat{s}_y, \hat{s}_z)$ is given by

$$\hat{\mathbf{X}} = [\hat{A}]\hat{\mathbf{X}}_0, \text{ where}$$

$$[\hat{A}] = \exp[\hat{\theta}, \hat{\mathbf{s}}] = [\hat{I}] + \sin\hat{\theta}[\hat{S}] + (1 - \cos\hat{\theta})[\hat{S}]^2, \qquad (7)$$

in which $[\hat{A}]$ is the displacement matrix determined by $\hat{\theta}$ and $[\hat{\mathbf{s}}]$, $[\hat{I}]$ is the dual identity matrix, and $[\hat{S}]$ is a skew-symmetric matrix given by

$$[\hat{S}] = \begin{bmatrix} 0 & -\hat{s}_z & \hat{s}_y \\ \hat{s}_z & 0 & -\hat{s}_x \\ -\hat{s}_y & \hat{s}_x & 0 \end{bmatrix}, \qquad (8)$$

which has the property that $[\hat{S}]\hat{\mathbf{S}} = \hat{\mathbf{s}} \times \hat{\mathbf{S}}$ for any screw $\hat{\mathbf{S}}$. See Figure 2 for illustration.

2.4.2 Constructing the Displacement Matrix with Two Parallel Control Screws

If the displacement is a pure translation, the initial and final locations of the screw are parallel. When the displacement time from $\hat{\mathbf{X}}_0$ to $\hat{\mathbf{X}}_1$ is $t \in [0, 1]$, the displacement matrix takes the special form

$$[\hat{A}] = [A] + \epsilon[\hat{D}][A], \qquad (9)$$

where $[A]$ is a rotation matrix and the distance between the two control screws embodies the translation of the line, $[\hat{D}]$ is a distance matrix, and $\hat{\mathbf{D}}$ is the dual perpendicular axis of the initial and final locations of the interpolated screws. The rotation matrix is the identity matrix for a parallel translation. The dual displacement matrix can then be constructed as

$$[\hat{A}] = [\hat{I}] + \epsilon t[\hat{D}]. \qquad (10)$$

It becomes

$$[\hat{A}] = [\hat{I}] + \epsilon[\hat{D}], \quad when \quad t = 1. \qquad (11)$$

For this situation, we have a special method to construct $\hat{\mathbf{D}}$. In Figure 3, $\mathbf{R}_1(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{Z}_1)$ and $\mathbf{R}_2(\mathbf{X}_2, \mathbf{Y}_2, \mathbf{Z}_2)$ are origin radius vectors of the control screws $\hat{\mathbf{X}}_0$ and $\hat{\mathbf{X}}_1$ separately. $\hat{\mathbf{X}}_t$ is a random position of the interpolated screw over $t \in [0, 1]$. The point r_1 is the intersection point of $\mathbf{R}_1$ and $\hat{\mathbf{X}}_0$; r_2 is the intersection point of $\mathbf{R}_2$ and $\hat{\mathbf{X}}_1$. Let

$$\hat{\mathbf{D}} = \mathbf{D} + \epsilon \mathbf{D_0}. \qquad (12)$$

The real part of $\hat{\mathbf{D}}$, denoted by $\mathbf{D}$, is the direction component of $\hat{\mathbf{D}}$. In Figure 3, we find that $\mathbf{D} = \mathbf{r}_2 - \mathbf{r}_1$. The moment of $\hat{\mathbf{D}}$ may be derived by the following formula

$$\mathbf{D_0} = \mathbf{R_1} \times \mathbf{D} = \mathbf{R_2} \times \mathbf{D}. \qquad (13)$$

After $\hat{\mathbf{D}}$ is derived, $[\hat{D}]$ can be easily constructed using the components of $\hat{\mathbf{D}}$. In a nutshell, $\hat{\mathbf{D}}$ can be calculated from the intersection points of $\hat{\mathbf{D}}$ and $\hat{\mathbf{X}}_0$, $\hat{\mathbf{X}}_1$, $\hat{\mathbf{X}}_t$ when $\hat{\mathbf{X}}_0$ and $\hat{\mathbf{X}}_1$ are parallel. $[\hat{D}]$ embodies the translation of the screw $\hat{\mathbf{X}}_t$.

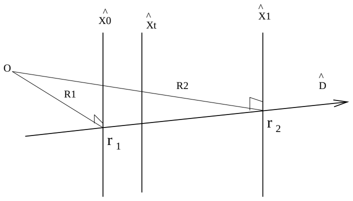


Figure 3. The relation between translation component $\hat{\mathbf{D}}$ and $\hat{\mathbf{X}}_0$, $\hat{\mathbf{X}}_1$, $\hat{\mathbf{X}}_t$ when $\hat{\mathbf{X}}_0||\hat{\mathbf{X}}_1$.

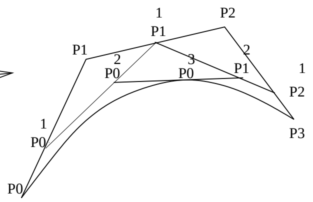


Figure 4. The de Casteljau algorithm for a third order Bezier Curve

3 Algorithms and Properties of Bezier Surfaces

3.1 The de Casteljau Algorithm

The (real space) de Casteljau algorithm is used to draw a *Bezier curve* by interpolation of several nominated control points, as shown in [6] and [4]. A set of $n+1$ control points is used to construct an n-th order polynomial curve. Given a set of control points P_0, P_1,... P_n, we can construct the curve

$$P_i^r(t) = (1-t)P_i^{r-1}(t) + tP_{i+1}^{r-1}(t), \tag{14}$$

where $P_i^0(t) = P_i$, $r = 1, \cdots, n$, and $i = 0, \cdots, n-r$. The point on the Bezier curve at $t \in [0,1]$ is given by $P_0^n(t)$. See Figure 4.

Theorem 1 *[2] The interpolation line is the tangential vector of its Bezier curve in the 2-dimensional space when all control points are on one side of the Bezier curve.*

3.2 The Dual de Casteljau Algorithm

The principle of dual space methods is to represent the operations on lines as becoming the same as a dualized version of spherical geometry [15].

The dual (space) de Casteljau algorithm [15] is derived from the (real space) de Casteljau algorithm by replacing the nominated points with screws (lines). The Bezier curve in the real method turns out to be the analogous ruled surface in the dual method.

For ruled surfaces, the displacement between two control screws must be chosen to follow a *geodesic* path on the *dual unit sphere* between those points which represent the unit lines $\hat{\mathbf{X}}_0^0$ and $\hat{\mathbf{X}}_1^0$. The basic procedure is as follows:

1. Find the dual angle between the two screws from

$$\hat{\theta} = \theta - \epsilon\frac{b}{\sin(\theta)}, \tag{15}$$

where $\theta = \cos^{-1}(a)$ and $a + \epsilon b = \cos\hat{\theta} = \hat{\mathbf{X}}_0^0 \cdot \hat{\mathbf{X}}_1^0$.

2. Find the screw axis for the motion from

$$\hat{\mathbf{s}} = \hat{\mathbf{X}}_0^0 \times \hat{\mathbf{X}}_1^0. \tag{16}$$

3. Find the displacement matrix $[\hat{A}]$ and the interpolated screw from

$$\hat{\mathbf{X}}_0^1 = [\hat{A}]\hat{\mathbf{X}}_0^0, \tag{17}$$

where $[\hat{A}] = [\hat{I}] + \sin(t\hat{\theta})[\hat{S}] + (1-\cos(t\hat{\theta}))[\hat{S}]^2$, $[\hat{I}]$ is the dual identity matrix, and $[\hat{S}]$ is a skew-symmetric matrix given by equation (7) and (8).

4. If the two control screws are parallel, the displacement matrix should be written as equation (10).

5. If there are $n+1$ control screws, we will get the n-th generation screws which constitute the ruled surface

$$\hat{\mathbf{X}}_i^r(t) = \exp(\hat{\theta}^{r-1}_{(i,i+1)}t[ad\hat{\mathbf{s}}^{r-1}_{(i,i+1)}])\hat{\mathbf{X}}_i^{r-1}, \tag{18}$$

where $\hat{\mathbf{X}}_i^0(t) = \hat{\mathbf{X}}_i$, $\hat{\theta}^{r-1}_{(i,i+1)}$ is the complex angle between ruled screws $\hat{\mathbf{X}}_i^{r-1}$ and $\hat{\mathbf{X}}_{i+1}^{r-1}$, $[ad]$ is an operator that takes a vector into a skew symmetric matrix, and $\hat{\mathbf{s}}^{r-1}_{(i,i+1)}$ is the line of the screw axis between $\hat{\mathbf{X}}_i^{r-1}$ and $\hat{\mathbf{X}}_{i+1}^{r-1}$. The ruled screws on the surface is given by $\hat{\mathbf{X}}_0^n(t)$.

The basic condition to draw a ruled surface using the dual de Casteljau algorithm is that the neighboring rulings of the ruled surface do not intersect each other except when the displacement is a pure rotation on a plane. Usually, there is the same fixed angle and distance between each pair of neighboring rulings. We denote this condition as (C-DDC) for short.

3.3 The Properties of Bezier Surfaces

For any operation defined for a real vector space there is a dual version with similar interpretations [17].

In this paper, ruled surfaces that are created by using the initial or variations of the dual de Casteljau algorithm are called Bezier surfaces. They have similar properties to the Bezier curves as in [6]. Points in the real space algorithm are normally replaced by screws (lines) in the dual space algorithm. Lines in the real space algorithm are replaced by planes in the dual space algorithm. The properties of Bezier (ruled) surfaces are investigated and listed below.

End ruling interpolation: the Bezier surface interpolated by a series of control screws (lines) does not generally pass through, or interpolate, all of the control screws, but it always interpolates (begins and ends on) the first and last ones.

Planar precision: if all the control screws are on the same plane, the Bezier surface will be “trapped” inside the plane and all the rulings in the ruled surface will be planar.

Affine invariance: the Bezier surface is invariant under affine parameter transformation. If we wish to transform a ruling on a Bezier surface to a new position using a transformation matrix and offset vector, we only need to transform the control screws and then use the same interpolation algorithm to create the transformed Bezier surface. Here, although the interpolation is dual, the affine transformation parameters are real. If we wish to transform $\hat{\mathbf{X}}(t)$ on to the new position $\hat{\mathbf{X}}'(t)$ using a real transformation matrix N and offset vector **tr**, we have

$$\hat{\mathbf{X}}'(t) = \hat{\mathbf{X}}(t)N + \mathbf{tr}. \tag{19}$$

Symmetry: for a control sequence of a Bezier surface, we can label $\hat{\mathbf{X}}_0$, $\hat{\mathbf{X}}_1$, $\cdots$, $\hat{\mathbf{X}}_n$ equivalently as $\hat{\mathbf{X}}_n$, $\hat{\mathbf{X}}_{n-1}$, $\cdots$, $\hat{\mathbf{X}}_0$. The surfaces that correspond to the two different orderings look the same. They only differ in the directions in which they are traversed. Thus, we have

$$\hat{\mathbf{X}}_j'(t) = \hat{\mathbf{X}}_{n-j}(1-t). \tag{20}$$

The above properties have been implicitly confirmed through different implementation examples in later sections of this paper. These properties are analogous to those of the real space algorithms in [6]. However we cannot prove that any real space properties of Bezier curves are analogous to their dual space properties of Bezier surfaces. There are fewer properties for Bezier surfaces than for Bezier curves.

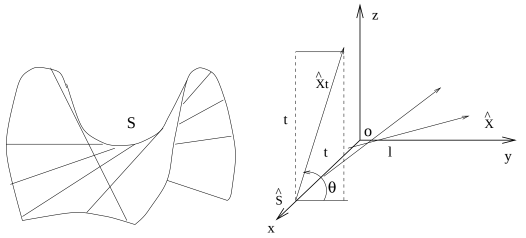


Figure 5. Geometrical illustration of the hyperbolic paraboloid.

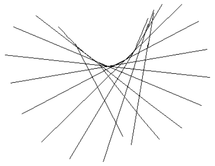


Figure 6. The ruled surface of the hyperbolic paraboloid is drawn by using the dual de Casteljau algorithm.

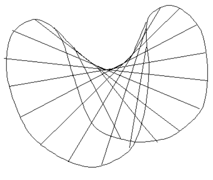


Figure 7. The skeleton of the hyperbolic paraboloid forms the interpolated surface into a hyperbolic paraboloid.

4 Drawing Particular Ruled Surfaces by Using the Initial Dual de Casteljau Algorithm

If the characteristics of a ruled surface meet the basic requirement of the dual de Casteljau algorithm (C-DDC) in section 3.1, the ruled surface is able to be drawn by using the initial dual de Casteljau algorithm. The hyperbolic paraboloid categorizes into this type of ruled surfaces.

4.1 The Hyperbolic Paraboloid

4.1.1 The Definition of the Hyperbolic Paraboloid

Carmo [1] defines the *hyperbolic paraboloid* as follows:- let S be the *hyperbolic paraboloid* $z = kxy$, $k \neq 0$. When S is a ruled surface, we observe that the lines $y = z/tk$, $x = t$, for each $t \neq 0$ belong to S. If we take the intersection of this family of lines with the plane $z = 0$, we obtain the curve $x = t$, $y = 0$, $z = 0$. Taking this curve as directrix and vectors $\omega(t)$ parallel to the lines $y = z/tk$, $x = t$, we obtain

$$\alpha(t) = (t, 0, 0) \text{ and } w(t) = (0, 1/k, t).$$

This gives the ruled surface illustrated in Figure 5:

$$x(t, v) = a(t) + vw(t) = (t, \frac{v}{\sqrt{1 + k^2t^2}}, \frac{vkt}{\sqrt{1 + k^2t^2}}),$$

where $t \in R$, $v \in R$, the trace of which clearly agrees with S.

Since $\alpha'(t) = (1, 0, 0)$, we deduce that the line of striction is α itself. The distribution parameter is $\lambda = 1/k$.

We also note that the tangent of the angle θ, which $\omega(t)$ makes with $\omega(0)$, is $tan\ \theta = tk$.

If the surface is intersected by a plane, which is parallel to the x-y plane, the resulting curve is a hyperbola. If the surface is intersected by a plane containing the z-axis, the resulting curve is a parabola.

4.1.2 The Dual Drawing Method of the Hyperbolic Paraboloid

Farin [4] used a real method:- a bilinear interpolation method to create the hyperbolic paraboloid. Now we create the hyperbolic paraboloid by using the dual de Casteljau algorithm. In Figure 5, we select as the original ruling a screw $\hat{\mathbf{X}}$ that is the y-axis. We select as the rotation axis an axis $\hat{\mathrm{s}}$ that is the x-axis. Here $\hat{\mathbf{X}} \perp \hat{\mathrm{s}}$. When $\hat{\mathbf{X}}$ rotates and translates along $\hat{\mathrm{s}}$ to the position $\hat{\mathbf{X}}_t$, where the rotation angle is $\theta < 90^o$, the hyperbolic paraboloid is created. However, we need to add a constraint between the rotation angle and the translation distance. When we know $\hat{\mathbf{X}}$, we find that axis $\hat{\mathrm{s}}$ has the relationship: $\hat{\mathrm{s}} \cdot \hat{\mathbf{X}} = 0$. Then $\hat{\mathrm{s}}$ intersects $\hat{\mathbf{X}}$ at some point O. For drawing the rulings, we set the translation distance as $d = tD$, where D is the distance between the final ruling $\hat{\mathbf{X}}_t$ and the first ruling $\hat{\mathbf{X}}$, for $t \in [0, 1]$. In this way, the i-th rotation angle will be $\theta_i = \arctan(tDk)$, where k is the distribution parameter. Thus, to create the hyperbolic paraboloid using the dual de Casteljau Algorithm, we take the following steps:

1. Set an origin screw $\hat{\mathbf{X}}$ that is the first ruling of the hyperbolic paraboloid surface.

2. Calculate the displacement axis $\hat{\mathrm{s}}$ using the relationship $\hat{\mathbf{X}} \cdot \hat{\mathrm{s}} = 0$, where $\hat{\mathrm{s}}$ and $\hat{\mathbf{X}}$ intersect at a common point.

3. Set the constraint between the translation distance d and the rotation angle θ_i, then

$$\tan\theta_i = dk. \tag{21}$$

Let $d = tD$, where D is the distance between the first and last rulings, and $t \in [0, 1]$. Equation (21) becomes ($0 \leq \theta_i < 90^o$)

$$\theta_i = \arctan(tDk). \tag{22}$$

4. The dual angle of the displacement will become

$$\hat{\theta}_i = \theta_i + \epsilon d = \arctan(tDK) + \epsilon(tD). \tag{23}$$

5. The displacement matrix is

$$[\hat{A}_i] = [\hat{I}] + \sin(\hat{\theta}_i)[\hat{S}] + (1 + \cos(\hat{\theta}_i))[\hat{S}]^2. \tag{24}$$

6. The interpolated screws will be

$$\hat{\mathbf{X}}_i = [\hat{A}_i]\hat{\mathbf{X}}. \tag{25}$$

The size and shape of the hyperbolic paraboloid are determined by $\hat{\theta}_i$, which is in turn determined by k and D. If k increases while D is a constant, then the distance between neighbouring rulings will increase and the shape of the hyperbolic paraboloid will widen. If k decreases, then the distance between neighbouring rulings will decrease and the shape of the hyperbolic paraboloid will narrow down. If D increases while k is a constant, then the shape of the hyperbolic paraboloid will widen. Otherwise, it becomes narrower.

In Figure 5, we let $\hat{\mathbf{X}}$ overlap axis $\mathbf{x}$. Note that $\hat{\mathrm{s}}$ is axis $\mathbf{y}$, and that $\hat{\mathbf{X}}$ displaces along axis $\hat{\mathrm{s}}$ in the same $\hat{\theta}$ as in the above method. Then, we have a reciprocal hyperbolic paraboloid of the original one.

The figures 6 and 7 are the computer implementations of the dual drawing method for the hyperbolic paraboloid.

5 Drawing Particular Ruled Surfaces by Using Variations of the Dual de Casteljau Algorithm

If the characteristics of a ruled surface do not meet the basic requirement of the dual de Casteljau algorithm (C-DDC

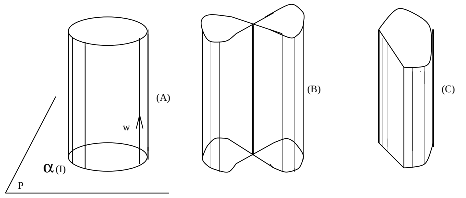


Figure 8. The different shaped cylinders

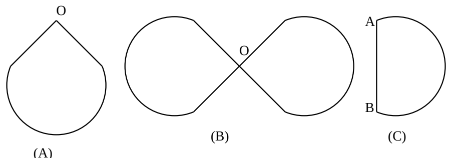


Figure 9. The different shaped base curves of cylinders

for short) in section 3.1, the ruled surface is not able to be drawn by using the initial dual de Casteljau algorithm. Variations and additional conditions of the dual de Casteljau algorithm are explored in order to draw particular ruled surfaces:- cylinders, cones and hyperboloids.

5.1 Cylinders

5.1.1 Regular Cylinders

5.1.1.1 The Definition and Analysis of the Cylinder

As defined in [1], a *cylinder* is a ruled surface generated by a one-parameter family of lines $\{\alpha(t), \omega(t)\}$, $t \in I$, where $\alpha(I)$ is contained in a plane P and $\omega(t)$ is parallel to a fixed direction in R^3 (see Figure 8(A)).

Technically, cylinders can be created with the dual de Casteljau algorithm because they meet the preconditions of the algorithm (C-DDC) in section 3.1. However, for cylinders with smooth trajectories, we need to add certain boundary conditions to deal with smoothness when two end control screws of a smooth trajectory overlap each other. Setting all control screws (lines) parallel to each other is the basic way to create any cylindrical surface with the dual de Casteljau algorithm.

The base curve of a cylinder could be an open or closed curve. Closed curves may be classified into two kinds. One kind is the smooth closed curve which has a first derivative at any point on the curve; e.g., the circle. The other is the non-smooth closed curve without first derivatives at vertex points on the curve, such as the shapes shown in Figure 9.

5.1.1.2 Cylinders Based on Open or Non-smooth Closed Trajectories

From [6], a Bezier curve begins at the first control point and ends at the last control point. The two points can be two ends of an open curve. It was found in Section 3.3 that they are also suitable for the dual space case. That is, a ruled surface based on a series of control screws always interpolates the first and last control screws.

If the base curves of cylinders are open curve trajectories , we could create cylinders by using the dual de Casteljau algorithm directly. Thus, we focus on the drawing methods of cylinders based on closed curve trajectories.

We break down a non-smooth closed curve into several different curve segments. For example, in Figure 9, curve (B) can be broken into two pieces of (A). We also think of them as open curve segments, which join at the non-smooth intersection points. For example, in Figure 9(A), the closed curve is actually an open curve connected at point O. The curve in Figure 9(C) actually consists of two open curves:- an arc, and a straight line connected at point A and B. We draw the non-smooth closed curve as the open Bezier curve using the real de Casteljau algorithm straight away (Figure 9(C)). Creating cylinders along these non-smooth closed curves with the dual de Casteljau algorithm is analogous to using the real space drawing method for the non-smooth closed curves, but replacing the control points in the real space method with the dual parallel screws (lines) in the dual space method.

5.1.1.3 The Dual Drawing Method of The Cylinder Based on a Smooth Closed Trajectory

It is supposed that the smooth closed trajectory is split into a curve with starting and end points at the same point on the curve. Thus a smooth closed curve can be seen as an open curve with two end points on the same point. The cylinder based on this base curve can be drawn by using the dual de Casteljau algorithm. In order to guarantee that the base curve and the cylinder are smooth, we have to consider that the overlap end point on the curve has a tangent vector and that the analogous overlap ruling on the cylinder has a tangent plane. Thus, we should determine the boundary conditions to draw the cylinder with the smooth closed curve trajectory by using the dual de Casteljau algorithm.

We first transfer the dual space ruled surface issue to a real space curve issue to simplify the problem. The cylinder is constructed by a series of parallel rulings along a base curve. The rulings are supposed to be created by using the dual de Casteljau algorithm with parallel control screws. We consider that the base curve of the cylinder is projected to the reference plane that is perpendicular to the rulings of the cylinder and their control screws. The projected curve is actually a Bezier curve as in Figure 12. The purpose of this conversion is to try to find the boundary conditions if two end points of a Bezier curve join together smoothly on the reference plane. Once the boundary condition of the join point of the smooth curve is found, the boundary connection of a smooth ruled surface of a cylinder is also solved by analogizing the curve boundary condition.

For a Bezier curve on a reference plane, if the curve segments of the Bezier curve join each other smoothly, the first derivative at the join point should exist. Equivalently there should be only one tangent vector at the join point.

For the ruled surface of a cylinder, there must be a tangent plane on the joint ruling of two connected ruled surfaces if the joint ruled surface is smooth.

In Figure 14, The first control point P_0 is on the line L_1, which is determined by the first and second control points, and the last control point P_n is on the line L_n, which is determined by the last and second to last control points. When $t = 0$, for the de Casteljau algorithm, the first interpolated line at any level is identical to L_1(Figure 10); when $t = 1$, the last interpolated line at any level is identical to L_n (Fig-

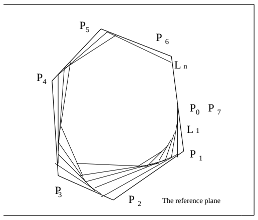


Figure 10. When $t = 0$, the interpolated line between the first and second control points at any level are identical to the same position interpolated line at the previous level

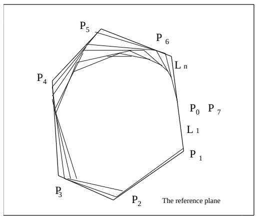


Figure 11. When $t = 1$, the interpolated line between the last and second to last control points at any level are identical to the same position interpolated line at the previous level

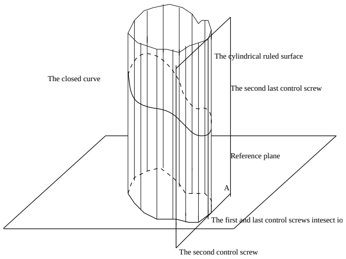

The closed curve
The cylindrical ruled surface
The second last control screw
Reference plane
A
The first and last control screws intesect io
The second control screw

Figure 12. The first and last control screws overlap on plane A which is perpendicular to the reference plane

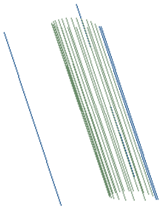


Figure 13. The computer implementation of the cylinder using the variation of the dual de Casteljau algorithm

ure 11). From theorem 1, it can be proved that L_1 and L_n are the tangent vectors of the curve at its first and last points separately. If we want to draw the smooth closed curve using the real de Casteljau algorithm, we only need L_1 and L_n to be linearly identical. When the first and second control points, and the last and the second to last control points, lie on the same line, the tangent vectors of the curve at the first and last control points will overlap. Thus the curve is connected on the end points smoothly because the tangent vector exists on the boundary points. Similarly in the dual issue, if the first, second, last and second to last control screws are on the same plane, the interpolated surface will have the same tangent plane on the first and last control screws. The edges of the ruled surface connect to each other smoothly because the tangent plane exists on the edge. Figure 12 illustrates the principle of drawing methods. The computer implementation of the cylinder is shown on Figure 13.

5.1.2 The Circular Cylinder

Practically, the positions of control screws of the dual de Casteljau algorithm are not easily determined for drawing certain shapes such as the circular cylinder. Piegl and Tiller [11] gave the Rational B-spline method for drawing a circle by inserting appropriate knots, which are the intersection points of the circle and its tangent square edge. We will determine the appropriate control screws to draw the circular cylinder. We will also observe whether the number and the positions of control screws determine the circular cylinder.

5.1.2.1 Drawing the Circular Cylinder Using the Dual de Casteljau Algorithm with Boundary Condition

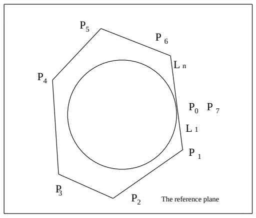


Figure 14. View of the cylinder section along the normal vector of the reference plane

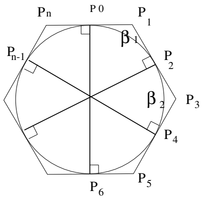


Figure 15. Creating a circle with the segment construction method

We look at the base Bezier curve of the circular cylinder (a circle) first in order to identify the positions of control points of the base circle, which are also those of control screws of the circular cylinder. In Figure 14,if the cylinder section is a circle,the condition of drawing the circle by using the de Casteljau algorithm is that

$$x^2 + y^2 = 1. \tag{26}$$

Here, P_0, P_1, P_{n-1} and P_n are co-linear, and $P_0 = P_n(e.g., P_7)$. We need five minimum control points to draw a circle. That is, besides P_0, P_1, P_{n-1} and P_n, there should be at least one more control point. The functional expression of the de Casteljau algorithm for drawing a circle as in [4] is

$$P(t) = P_0g_0(t) + P_1g_1(t) + \cdots P_kg_k(t) + \cdots + P_ng_n(t)$$

$$= \sum_{k=0}^{n} P_k g_k(t) = \sum_{k=0}^{n} P_k B_k^n(t),$$

where $g_k(t)$ is the Bernstein polynomial and

$$g_k(t) = B_k^n(t) = \tbinom{n}{k}(1-t)^{n-k}t^k, \tag{27}$$

$P_k = (P_{kx}, P_{ky})$ is the control point of the circle, and $P(t)$ is the point on the circle with coordinate $(P_x(t), P_y(t))$ when $t \in [0, 1]$. Then,

$$\begin{aligned} P(t) &= \sum_{k=0}^{n} P_k g_k(t) = \sum_{k=0}^{n}(P_{kx}, P_{ky})g_k(t) \\ &= \sum_{k=0}^{n}(P_{kx}g_k(t), P_{ky}g_k(t)) \\ &= \left(\sum_{k=0}^{n} P_{kx}g_k(t), \sum_{k=0}^{n} P_{ky}g_k(t)\right) = (P_x(t), P_y(t)), \end{aligned}$$

where $P_x(t)$ and $P_y(t)$ satisfy the circle condition from equation (26). Thus,

$$\begin{aligned} (P_x(t))^2+(P_y(t))^2 &= \left(\sum_{k=0}^{n} P_{kx}g_k(t)\right)^2 + \left(\sum_{k=0}^{n} P_{ky}g_k(t)\right)^2 \\ &= (P_{0x}g_0(t) + P_{1x}g_1(t) + \cdots + P_{nx}g_n(t))^2 \\ &\quad +(P_{0y}g_0(t) + P_{1y}g_1(t) + \cdots + P_{ny}g_n(t))^2 \\ &= 1. \end{aligned} \tag{28}$$

We must also constrain the circle by the condition of P_0, P_1, P_{n-1}, and P_n being co-linear, and $P_0 = P_n$.

Solving the above equations to find the control points, we need to know the first and last control points P_0, P_n in advance. The second and second last control points will be determined on the line $P_0P_1P_{n-1}P_n$ by using the co-linear constraint condition. Then we use the different values of $t \in [0, 1]$

in equations (27) and (28) to find other control points, and the number w of chosen constants t should be less than twice the number $n+1$ of control points, (i.e.,$w < 2(n+1)$), to guarantee that the equations are solvable and yield the coordinates of control points. If P_1 and P_{n-1} are not chosen well, we select other positions similarly as P_1 and P_{n-1} on the line $P_0P_1P_{n-1}P_n$ until all control points are solvable from the equations.

We replace the control points of the base circle with parallel screws (lines), and let all these control screws be perpendicular to the plane where the circle lies. The circular cylinder is then created by using the dual de Casteljau algorithm based on these control screws.

5.1.2.2 Drawing the Circular Cylinder with Segment Construction Using the Dual de Casteljau Algorithm

We first consider the circle segment, which is the base curve of the circular cylinder segment, in order to identify the positions of control points of the base curve and also the control screws of the circular cylinder segment. We use the de Casteljau algorithm to create curve segments which satisfy the Bezier curve, then connect them as a whole circular curve. Similar rules are applied to constructing the circular cylinder by their segments drawn with the dual de Casteljau algorithm.

In Figure 15, the curve segment P_0P_2 is created by control points P_0, P_1 and P_2. Similar methods apply to other segments. The P_0P_2 curve is weighted by the control points P_0, P_1, and P_2 only. The P_2P_4 curve is weighted by the control points P_2, P_3, and P_4 only, and so on. To draw the next circular segment, the positions of the control points need to be found by using a similar method as the previous section.

We also find that the intersection points of the tangent polygon of the circle and the circle itself are the boundary points of curve segments. The radii of the circle are perpendicular to their relative polygon edges when the radii intersect at the boundary points of curve segments. For example, in Figure 15, P_0P_6 is one of the diameters of the circle, and P_0 and P_6 are both boundary points of the curve segments, where $P_0P_6 \perp P_1P_n$, $P_0P_6 \perp P_5P_6$.

We suppose that there are $n+1$ control points from P_0, ..., P_n in total for a circle, which is constructed by m circle segments in $t \in [0, t_{1f}, t_{2f}, \ldots, t_{kf}, \ldots, t_{mf}]$, and each circle segment is constructed by $j+1$ control points in $t \in [t_{(k-1)f}, t_{kf}]$. We have $m \cdot J = n+1$.

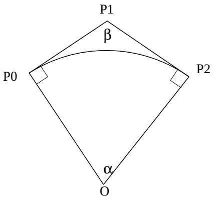


Figure 16. The circular cylinder segment drawing method

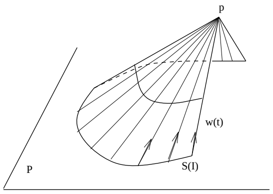


Figure 17. The explanation of the definition of cones.

For these curve segments, their functional expression of the de Casteljau algorithm is

$$S_1(t_1) = P_0g_0(t_1) + P_1g_1(t_1) + P_2g_2(t_1) + \cdots + P_jg_j(t_1),$$

$$S_2(t_2) = P_jg_j(t_2) + P_{j+1}g_{j+1}(t_2) + P_{j+2}g_{j+2}(t_2) +$$
$$\cdots + P_{2j}g_{2j}(t_2),$$
$$\cdots$$
$$S_k(t_k) = P_{(k-1)j}g_{(k-1)j}(t_k) + P_{(k-1)j+1}g_{(k-1)j+1}(t_k)$$
$$+ \cdots + P_{kj}g_{kj}(t_k),$$
$$\cdots$$
$$S_m(t_m) = P_{(m-1)j}g_{(m-1)j}(t_m) + P_{(m-1)j+1}g_{(m-1)j+1}(t_m)$$
$$+ \cdots + P_{mj}g_{mj}(t_m),$$

where $S_1(t_1)$ is the point on the first circle segment at $t_1 \in [0, t_{1f}]$, $S_k(t_k)$ is the point on the k^{th} circle segment at $t_k \in [t_{(k-1)f}, t_{kf}]$ and $S_m(t_m)$ is the point on the m^{th} circle segment at $t_m \in [t_{(m-1)f}, t_{mf}]$. Then, the whole circle is constructed by

$$P(t) = S_1(t_1) + ... + S_m(t_m) \;\; in \;\; t \in [0, t_{mf}].$$

We take the first segment as in the example under the same supposed conditions, and specify $j+1=3$. In Figure 16, we suppose that control point $P_0(P_{0x}, P_{0y})$ is known. Control points $P_1(P_{1x}, P_{1y})$ and $P_2(P_{2x}, P_{2y})$ are not known. As equation (28), the curve segment satisfies the circle condition

$$(P_{0x}g_0(t_1) + P_{1x}g_1(t_1) + P_{2x}g_2(t_1))^2$$
$$+(P_{0y}g_0(t_1) + P_{1y}g_1(t_1) + P_{2y}g_2(t_1))^2 = 1, \qquad (29)$$

where $g_i(t_1)$ are Bernstein polynomials and determined by $t_1 \in [0, t_{1f}]$ for the first curve segment, but $t_k \in [t_{(k-1)f}, t_{kf}]$ in general. The number w_k of chosen t_k should be less than double the number $(j+1)$ of control points. That is, $w_k < 2(j+1)$. Thus it guarantees that the equations are solvable to derive the positions of control points P_1 and P_2. However, if the quotient of the angle α at the centre dividing $360°$ is not an integer, the circle curve segments will not join as a complete circle by this method only. Thus, before deducing the circle segments, we need to constrain the quotient to be an integer. In Figure 16, the curve (arc) angle is α, $P_0O \perp P_0P_1$, and $P_2O \perp P_1P_2$. It is easy to get the angle β from P_0, P_1, and P_2 because the positions of P_1 and P_2 from equation (29) have already been calculated and P_0 is known. In the following equations, we have

$$\alpha + \beta = 180°, m = \frac{360°}{\alpha} = \frac{360°}{180 - \beta}. \qquad (30)$$

If m is an integer, then α and β do not need to be adjusted. Otherwise, we have to adjust the position of $P_1$and P_2 until the angle β and α satisfy the constraint so as to make m an integer. Then m determines the number of circle segments. If the angle α does not satisfy the angle constraint, we adjust the position P_1 on line P_0P_1, then re-calculate P_2 by equation (29) and check whether the angle constraint is satisfied. We repeat the above procedure until the final angle condition is satisfied.

After we get the positions of control points of the first circle segment, we can get those of the next segment easily. In Figure 15, we must find whether P_3 is on the line P_1P_2, while $P_2P_3 = P_1P_2$, angle $\beta_2 = \beta_1$, and $P_2P_3 = P_3P_4$. Points P_2, P_3, and P_4 will be the control points of the second circle segment. From this, the 3^{rd}, 4^{th}, and all of the other control

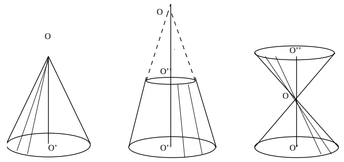


Figure 18. Different conic shapes based on circles.

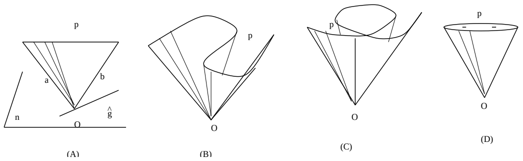


Figure 19. Various cones based on different curves.

points will be interpolated smoothly. These circle segments construct the whole circle.

For the circular cylinder, we replace the control points of the base circle segment, a Bezier curve, with parallel control screws which are perpendicular to the base plane. A related circular cylinder segment is then created by using the dual de Casteljau algorithm. One after another, the above method is applied to all circular cylinder segments with a same base plane. Ultimately, the whole circular cylinder is constructed by the circular cylinder segments.

5.2 Cones

5.2.1 The Analysis of Cones

5.2.1.1 The Definition, Classifications and Common Characteristics of Cones

Carmo [1] defines the *cone*. A *cone* is a ruled surface generated by a family $\{s(t), \omega(t)\}$, $t \in I$, where the curve $s(I)$ resides in a plane P and the rulings L_t all pass through a point $p \notin P$. See Figure 17.

Figure 18 illustrates the common characteristics of cones. In Figure 18, a cone in (A) has an explicit vertex O. In (B) of the same figure, extension lines of rulings of a cone intersect at a vertex which is implicit. In (C) of the same figure, the intersecting vertex of a cone is at point O. Thus, the common characteristic of the cone is that the rulings, or the extension rulings of the cone, always intersect at one point.

If the base curve of a cone is a circle, the normal vector of the base plane from the centre of the circle usually goes through the vertex of the cone. These cones with circular base curves shown in Figure 18 are called circular cones.

Normally cones are not necessarily based on circles. If we displace a straight line k passing through a moving point $p \in S(I)$ and a fixed point $O \notin S(I)$, we can create a cone with any base curve other than a circle. Figure 19 illustrates different conic shapes.

5.2.1.2 The Analysis of Creation Methods of Cones

Although cones are ruled surfaces, they can not be created with the dual de Casteljau algorithm directly. Because the rulings of cones intersect at one point, they do not obey the basic rule of the dual space algorithm (C-DDC for short) in section 3.2.

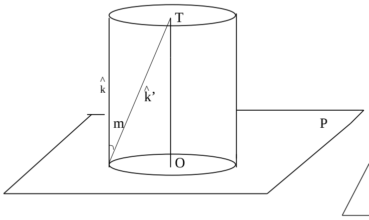


Figure 20. The relationship of rulings of a cylinder and rulings of a cone with a common base curve.

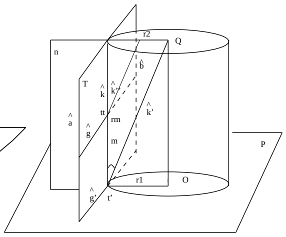


Figure 21. The dual method to convert a circular cone from a circular cylinder.

In Figure 20, we find that a circular cone can be converted from a circular cylinder. The ruling of the cylinder $\hat{\mathbf{k}}$ rotates along some axis to the ruling of the cone $\hat{\mathbf{k}}'$, which intersect the normal vector OT of the base plane at vertex T. Repeating the process to convert rulings of the whole cylinder, the cone is thus created. Now we must derive a displacement method to implement the conversion from the cylinder to the cone.

The circular cylinder has been created using the dual space methods in section 5.1.2. Therefore we only need to find a way to rotate rulings of the cylinder to rulings of the cone using a displacement method. Because we already know that the ruling of the cylinder $\hat{\mathbf{k}}$ rotates with angle m about some axis to the position $\hat{\mathbf{k}}'$ of the ruling of the cone, the key problem becomes to find an axis along which the screw $\hat{\mathbf{k}}$ rotates.

We suppose that the base curve of a Bezier surface such as the cylinder is created under 1. Thus, the interpolation line of the base curve could be the axis along which the ruling of a cylinder rotates to the ruling of a cone with the same base curve.

Based on theorem 1, we have the general deduction as follows:- the tangential interpolation line of a Bezier curve in the real space method becomes a tangential plane of a Bezier surface in the dual space method. Also, the tangential interpolation line of the base curve of a Bezier surface is the common perpendicular axis of two parallel control screws. Both results of this deduction are constrained by the condition that all control screws of a Bezier surface, such as the cylinder, are parallel. We denote this deduction as (TI-DD) for short.

5.2.2 The Dual Space Creation Methods of Cones

5.2.2.1 The Dual Space Drawing Methods of Circular Cones

Figure 20 is converted to Figure 21 to further display the drawing methods of cones. In Figure 21, the interpolation axis $\hat{\mathbf{g}}$ and its perpendicular control screws $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ are on the same plane T, which is the tangential plane of the interpolated cylinder on $\hat{\mathbf{k}}$ when $t = t_1, t_1 \in [0, 1]$. The interpolated ruled screw $\hat{\mathbf{k}}$ of the cylinder and its two control parallel screws $\hat{\mathbf{a}}$, $\hat{\mathbf{b}}$ are also planar on plane T. The cylinder should not be in the situation that all its control screws are planar, otherwise it transforms into a plane. The transformation axis $\hat{\mathbf{g}}$ of control screws $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ intersects the cylinder at only one point t on screw $\hat{\mathbf{k}}$, because any interpolation axis intersects the interpolated ruled line of a Bezier curve at one point and also because $\hat{\mathbf{g}}$ is a tangential vector of the cylinder according to the deduction (TI-DD) in the previous section. For the normal control screws, the position of the displacement axis $\hat{\mathbf{g}}$ is determined by the shortest distance between two control screws. Because control screws $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ are parallel to each other, it follows that the distance between them is the same everywhere. It is possible to move the axis $\hat{\mathbf{g}}$ to the position $\hat{\mathbf{g}}'$. This does not change the creation of the ruled surface of the cone since $\hat{\mathbf{g}}'||\hat{\mathbf{g}}$, and $\hat{\mathbf{g}}'$ is also on the plane T when $\hat{\mathbf{a}}||\hat{\mathbf{b}}$. Everything on plane P can be explained by the real space method. Thus, the projection points of $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ to plane P become the control points of the base Bezier curve $\odot O$ of the cylinder. $\hat{\mathbf{g}}'$ is the tangential interpolation line of the Bezier curve ($\odot O$) at point t' on plane P according to theorem 1.

In Figure 21, we create plane n which is perpendicular to plane T, and passes through screw $\hat{\mathbf{k}}$ and line OQ (because r_1 is the radius of $\odot O$; r_2 is the radius of $\odot Q$; note that $r_1||r_2$. r_1 and r_2 both are perpendicular to plane T). The interpolated ruled screw $\hat{\mathbf{k}}$ and the central line of the cylinder OQ are planar in plane n. Then $\hat{\mathbf{k}}$ is rotated with angle m to position $\hat{\mathbf{k}}'$ about axis $\hat{\mathbf{g}}'$. In plane n, $\hat{\mathbf{k}}'$ is one of the ruled screws of the cone and touches Q. If we vary t over $[0, 1]$ when we perform the process, the ruled surface of the cone is created. The method for rotating the screw $\hat{\mathbf{k}}$ to $\hat{\mathbf{k}}'$ is much simpler than we might think, because $\hat{g}'$ is perpendicular to plane n and is known as the interpolation axis between control screws $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$. Let $\hat{\theta} = \theta + \epsilon d$, where $d = 0$, and angle $\theta = m$. Equation (17) will become $\hat{k}' = [\hat{A}]\hat{k}, \quad where$

$$[\hat{A}] = [\hat{I}] + \sin(m)[\hat{g}'] + (1 - \cos(m))[\hat{g}']^2. \qquad (31)$$

This is the pure rotation equation to convert the rulings of a circular cylinder into the rulings of a circular cone. The angle m becomes constant rather than t dependant as in equation (17). The axis $\hat{\mathbf{g}}$ varies when the positions of the previous level control screws change through different $t \in [0, 1]$.

Alternatively, in Figure 21, we use the above method to rotate $\hat{\mathbf{k}}$ with angle m about axis $\hat{\mathbf{g}}$ to $\hat{\mathbf{k}}''$ first, then we move $\hat{\mathbf{k}}''$ to position $\hat{\mathbf{k}}'$. Since $\hat{\mathbf{k}}''||\hat{\mathbf{k}}'$, it is easily done. Thus, the rulings of the cone is also converted from that of the cylinder.

The shape of the circular cone is determined by the angle m and the basic circle radius. When we draw general circular cones, we only need to adjust the angle m, or the radius of the circular cylinder.

The above method applies to any circular conic shape, as shown in Figure 18. In some cases, however, we need to clip rulings of conic surfaces at different positions. In Figure 18(B), we extend the rulings above the top circle. In Figure 18(C), we clip the ruled screws on the top circle. When the base curve of the cone deteriorates into a line (such as in Figure 19(A)), it is taken as a special case of the conic shape. The pure rotation of the rulings about axis $\hat{\mathbf{g}}$ from a to b between $t \in [0, 1]$ will fulfil the task, when $\hat{\mathbf{g}}$ is perpendicular to the plane n. Figure 23 is the computer implementation of the method above.

5.2.2.2 The Dual Space Drawing Method of Cones With Random Shapes

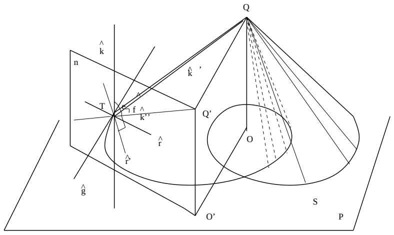


Figure 22. The dual space method to create random conic shapes.

We now investigate how to create conic surfaces that are based on more general curves such as those in Figure 19(B) and (C). In these situations, the use of angle m becomes more complicated. The ruled screw of the conic shape is not a pure rotation about some fixed axis from the ruled screw of the cylinder anymore. It contains some translation or another rotation– multi-translation and multi-rotation. Although the rotation axis is still the tangential vector to the related cylinder of the cone, the vector obtained from the ruling of the cylinder rotating along the axis is no longer the ruling of the cone as it was in the circular cone. As shown in Figure 22, after a rotation of angle m about the interpolation axis $\hat{\mathbf{g}}$ from relative cylinder ruling $\hat{\mathbf{k}}$ to $\hat{\mathbf{k}}''$, screw $\hat{\mathbf{k}}''$ is not on the position of the ruling of the cone and is not planar with the center axis of the conic shape OQ as it was in the creation method of the circular cone. The ruling $\hat{\mathbf{k}}'$ of the cone is not identical to $\hat{\mathbf{k}}''$ which is rotated from $\hat{\mathbf{k}}$. To solve these problems, a detailed method to create any shaped cone will be presented.

In Figure 22, the ruled screw of the conic shape $\hat{\mathbf{k}}'$ passes through point Q to construct the ruled conic surface along curve S. But the perpendicular vector $\hat{\mathbf{r}}$ of the rotation axis $\hat{\mathbf{g}}$ at point T does not necessarily intersect line OQ at the central point O of the base curve S as it does in the circular cone. There is no way to construct a plane that always makes screw $\hat{\mathbf{r}}$ and OQ planar. The standard dual drawing method of the circular cone cannot be used here directly. We need to modify it to suit this situation. Using the base curve in Figure 22, we draw a cylindrical surface with the dual de Casteljau algorithm as the methods in section 5.1.1. We do not show the cylindrical surface on the figure. $\hat{\mathbf{k}}$ is one of the rulings of this surface. We draw the plane n that is perpendicular to the interpolation screw $\hat{\mathbf{g}}$, intersects $\hat{\mathbf{g}}$ at T, and contains $\hat{\mathbf{r}}$. Then the screw $\hat{\mathbf{k}}$ can be rotated about interpolation axis $\hat{\mathbf{g}}$ with angle e to the position $\hat{\mathbf{k}}''$. $\hat{\mathbf{k}}''$ touches Q'. We draw line $Q'O'$ perpendicular to base plane P which contains the curve S. Note that $Q'O'||QO$. When $\hat{\mathbf{k}}$ rotates in angle e, we let screw $\hat{\mathbf{r}}$ rotate in angle e to $\hat{\mathbf{r}}'$ about axis $\hat{\mathbf{g}}$ as well. (Because $\hat{\mathbf{r}} \perp \hat{\mathbf{g}}$, $\hat{\mathbf{r}}$ is easy to be found using axis $\hat{\mathbf{g}}$ to rotate.) This makes $\hat{\mathbf{r}}' \perp \hat{\mathbf{k}}''$.

However, we need to make the plane TQQ' to be perpendicular to $\hat{\mathbf{r}}'$ in order to guarantee that $\hat{\mathbf{k}}''$ will rotate along axis $\hat{\mathbf{r}}'$ to the position TQ. To make $\hat{\mathbf{r}}'$ perpendicular to plane TQQ', we need to make $QQ'||\hat{\mathbf{g}}$. Because $\hat{\mathbf{r}}' \perp \hat{\mathbf{g}}$, we have $\hat{\mathbf{r}}' \perp QQ'$, when $\hat{\mathbf{r}}'$ is perpendicular to two lines QQ' and $\hat{\mathbf{k}}''$ which intersect each other, we have $\hat{\mathbf{r}}'$ perpendicular to the plane TQQ'. Now we only need to rotate $\hat{\mathbf{k}}''$ with angle f about axis $\hat{\mathbf{r}}'$ to the intersect vertex Q on the plane TQQ' to get the ruling of the conic shape TQ. In this way, the rotation axes $\hat{\mathbf{g}}$ and $\hat{\mathbf{r}}$ are easily calculated. Then $\hat{\mathbf{g}}$ is the axis of relative parallel control screws and can be obtained from the dual de Casteljau algorithm. The rotations are easily executed using the dual space method from equation (17).

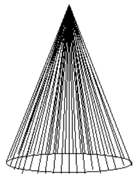


Figure 23. The circular cone drawn by using the dual space conversion method.

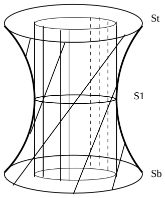


Figure 24. The relation between the cylinder and its created hyperboloid.

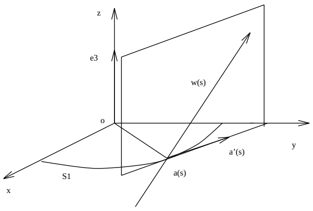


Figure 25. The real space geometrical explanation of the ruling of the hyperboloid.

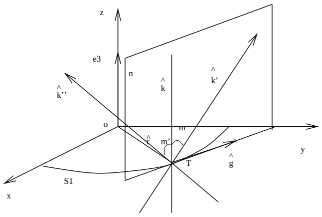


Figure 26. The dual space explanation of the ruling of the hyperboloid.

5.3 The Hyperboloid

5.3.1 The Definition of the Hyperboloid

Carmo [1] defined the *hyperboloid*. let S_1 be the unit circle $x^2 + y^2 = 1$ in the $x - y$ plane, and let $\alpha(s)$ be a parametrization of S_1 by arc lengths. For each s, let $\omega(s) = \alpha'(s) + e_3$, where e_3 is the unit vector of the z axis (See Figure 25). Then $X(s, v) = \alpha(s) + v(\alpha'(s) + e_3)$is a ruled surface. It can be written into the following more familiar form $X(s, v) = (\cos s - v \sin s, \sin s + v \cos s, v)$.Notice that $x^2 + y^2 - z^2 = 1 + v^2 - v^2 = 1$. This shows that the trace of X is a hyperboloid of revolution. See Figure 24.

It is interesting to observe that when we take $\omega(s) = -\alpha'(s) + e_3$, we again obtain the same surface. This shows that the hyperboloid of revolution has two sets of rulings.

5.3.2 The Dual Space Drawing Method of the Hyperboloid

Technically, the hyperboloid can be drawn by using the dual de Casteljau algorithm directory because it meets the basic requirement of the dual de Casteljau algorithm (C-DDC for short) in section 3.2. However, there is no way to control the size and shape of the hyperboloid. The size and shape of the hyperboloid are determined by its base circles which are on the middle, top and bottom of the hyperboloid. We note that the rulings of the hyperboloid are the rulings of a relative circular cylinder that have been rotated through an appropriate angle. The dual space drawing method of the circular cylinder has been described in section 5.1.2. We need to find a method to convert the circular cylinder to the hyperboloid.

We look at Figure 24. The lean rulings of the hyperboloid pass through the smallest base curve - $\odot S_1$, which is half way between the top curve - $\odot S_t$ and the bottom curve - $\odot S_b$. We consider that the base cylinder for the creation of the hyperboloid surface should be based on the curve - $\odot S_1$ and is created with the method in section 5.1.2. Then, we try to rotate the ruling by a certain angle to a new position to satisfy the requirement of the hyperboloid.

We observe that the common base circle of the cylinder and the hyperboloid could be the Bezier curve in theorem 1. The tangent plane on which the ruling of the cylinder lays also contains the tangential interpolation vector of the base circle. The tangential vector and plane are all perpendicular to the radius of the base circle.

we convert Figure 25, which is in the real space explanation, to Figure 26. The latter is in the dual space explanation. In Figure 26, the plane n is the tangential plane of the cylinder based on $\odot S_1$ of Figure 24. The tangential plane and the cylinder intersect on the ruling $\hat{\mathbf{k}}$, so the radius $\hat{\mathbf{r}}$ of $\odot S_1$ is perpendicular to the plane n. Then $\hat{\mathbf{k}}$ is rotated to the position $\hat{\mathbf{k}}'$, which is the ruling of the hyperboloid and is still on the plane n. We only need to rotate $\hat{\mathbf{k}}$ by angle m to $\hat{\mathbf{k}}'$ about axis $\hat{\mathbf{r}}$ to create the ruling of the hyperboloid.

Now, we try to determine axis $\hat{\mathbf{r}}$. Since the plane n is the tangent plane of the $\odot S_1$ at point T , we draw an axis $\hat{\mathbf{g}}$ through point T, which is in the intersection line of the plane containing $\odot S_1$ and the plane n. Because $\hat{\mathbf{k}}$ is the ruling of the circular cylinder, $\hat{\mathbf{k}} \perp \hat{\mathbf{r}}$, $\hat{\mathbf{g}}$ and $\hat{\mathbf{r}}$ are planar, so $\hat{\mathbf{k}} \perp \hat{\mathbf{g}}$, $\hat{\mathbf{g}}$ is one of the final level interpolation axes of the cylinder in creating the particular ruling $\hat{\mathbf{k}}$. So, $\hat{\mathbf{g}}$ is easily created from the control screws of related cylinder using the dual de Casteljau algorithm. Then $\hat{\mathbf{k}}$ is a ruling of the related cylinder, it is easily created as well. Because $\hat{\mathbf{g}} \perp \hat{\mathbf{r}}$, $\hat{\mathbf{g}} \perp \hat{\mathbf{k}}$, there are two ways to create $\hat{\mathbf{r}}$ from $\hat{\mathbf{g}}$ and $\hat{\mathbf{k}}$. One way is to rotate $\hat{\mathbf{g}}$ about axis $\hat{\mathbf{k}}$ at 90° to the position $\hat{\mathbf{r}}$. The second way is to use the following equations to obtain $\hat{\mathbf{r}}$ (from $\hat{\mathbf{r}} \perp \hat{\mathbf{k}}$, $\hat{\mathbf{r}} \perp \hat{\mathbf{g}}$):

$$\hat{\mathbf{r}} \cdot \hat{\mathbf{k}} = 0, \hat{\mathbf{r}} \cdot \hat{\mathbf{g}} = 0, \ \ then \ \ \hat{\mathbf{k}} \times \hat{\mathbf{g}} = \hat{\mathbf{r}}. \tag{32}$$

We already get $\hat{\mathbf{g}}$ and $\hat{\mathbf{k}}$ when the circular cylinder is created. Thus, the above two calculations about $\hat{\mathbf{r}}$ are easily solved. After the method is executed for every ruling of the cylinder, the hyperboloid is formed.

The shape and size of hyperboloids are dependent on the radius of $\odot S_1$ and the angle m. On the plane n, we can rotate $\hat{\mathbf{k}}$ along $\hat{\mathbf{r}}$ to a new direction $\hat{\mathbf{k}}''$, in the symmetrical position of $\hat{\mathbf{k}}'$ about axis $\hat{\mathbf{k}}$. The method for creating $\hat{\mathbf{k}}''$ is the same as that applying to the creation of $\hat{\mathbf{k}}'$. If m and $\odot S$ remain unchanged, $m' = m$, the two hyperboloids are the same. It demonstrates that the hyperboloid of revolution has two sets of rulings as in the definition.

Figures 27 and 28 show the computer implementation of the hyperboloid by using the above method. Figure 27 shows the interpolated screws. After the interpolated screws are framed by the skeleton, it appears the shape shown in Figure 28 which matches the hyperboloid exactly.

6 Presenting the Clipped ruling Segment on the Screen

It is very important to represent screws as lines on the screen where the ruled surface is described by using the screw presentation method. Ding [3] used the line segment algorithm [6] to draw the infinitely long screws on the screen. However, the screws on the screen are still infinite lines. Intuitively, an infinite line must be converted into line segments to render a ruled surface on the screen. Ravani and Wang [14]

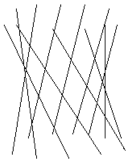


Figure 27. The ruled surface of the hyperboloid drawn by the dual space method.

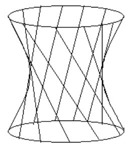


Figure 28. The ruled surface framed by the hyperboloid skeleton.

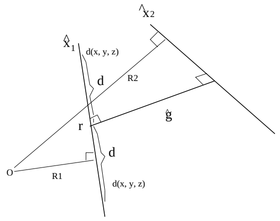


Figure 29. Drawing the clipped screw when $\hat{\mathbf{X}}_1$ is not parallel to $\hat{\mathbf{X}}_2$.

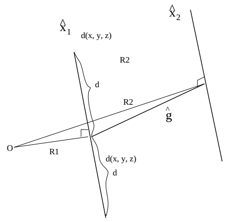


Figure 30. Drawing the clipped screw when $\hat{\mathbf{X}}_1$ is parallel to $\hat{\mathbf{X}}_2$.

proposed a method of representing line segments based on the "centre point" of a line. However, in the paper there is no way of controlling the location of the "centre point" during the interpolation. Ge and Ravani [5] proposed an intrinsic method based on the striction curve of the ruled surface. This method is coordinate frame independent, but there is little control over the location of the striction curve. Sprott and Ravani [15] proposed to select a reference point on the first control line and draw a reference line through the point. Then the reference line is interpolated along with the control screws using the same displacement matrix to yield the interpolated screws, in that the control screws and the reference line construct a rigid body (See [9]). The intersection point can be used to construct the boundary curve for the ruled surface patch. The line segment is yielded when one specifies a length along the line. Although this technique gives us some control over the location of the boundary curve, the task of setting the reference line is expensive in both computing time and design time. It cannot guarantee that the reference points is the central area of the interpolated screws as well.

We propose a method that shows the central area of the interpolated ruled surface on the screen. Because the characteristics of the ruled surface are presented in the control area the intersection points of the control screws and the common perpendicular axes determine the central area of the surface. The screw segments which we will draw are determined by these intersection points which we call the central points of the screw segments. The screw segments are augmented with two additional parameters representing the distance along the line on either side of the center point. The detailed method is stated as follows.

We review the algebraic form [12, 15] first. If a screw is, in i-frame with coordinates ($\hat{\mathbf{x}}$,$\hat{\mathbf{y}}$,$\hat{\mathbf{z}}$), $\hat{\mathbf{S}} \equiv \mathbf{S}_i + \epsilon \mathbf{S}_{pi}$, we can convert it to, in the real space system with coordinates $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, $\hat{\mathbf{S}} = \mathbf{S} + \epsilon \mathbf{S}_0$ using equation (5). The parametric form of converting a screw into a line in 3-dimensional space is

$$l(t) = \frac{\mathbf{S} \times \mathbf{S}_0}{\mathbf{S} \cdot \mathbf{S}} + t\mathbf{S}, \qquad (33)$$

where $\mathbf{R} = \frac{\mathbf{S} \times \mathbf{S}_0}{\mathbf{S} \cdot \mathbf{S}}$ is the point (foot) on the line that is closest to the origin of the chosen coordinate frame. t is any real number and selected with two or more different values. The screw will be drawn as a line in 3-dimensional space by connecting the points $l(t)$ yielded from different t values.

In Figure 29, we take one of two control screws $\hat{\mathbf{X}}_1$ as an example to show how to clip an infinite ruling (screw) into a segment. The two basic control screws are supposed $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$. The common perpendicular axis of $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ is $\hat{\mathbf{g}}$. Point O is the 3-dimensional coordinate origin. Also, d is the distance from the intersection point $r(r_x, r_y, r_z)$ of $\hat{\mathbf{g}}$ and $\hat{\mathbf{X}}_1$ to the selected point $d(x, y, z)$. Also, R_1, R_2 and R_g (R_g is not shown in the figure.) are origin radii of $\hat{\mathbf{X}}_1$, $\hat{\mathbf{X}}_2$ and $\hat{\mathbf{g}}$ respectively. We calculate the intersection point r of $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{g}}$. We suppose $\hat{\mathbf{X}}_1 = \mathbf{a} + \epsilon \mathbf{a}_0$ and $\hat{\mathbf{g}} = \mathbf{g} + \epsilon \mathbf{g}_0$, then we have $\mathbf{R}_1 = (\mathbf{a} \times \mathbf{a}_0)/(\mathbf{a} \cdot \mathbf{a})$ and $\mathbf{R}_g = (\mathbf{g} \times \mathbf{g}_0)/(\mathbf{g} \cdot \mathbf{g})$ to calculate $\mathbf{R}_1$ and $\mathbf{R}_g$. From equation (33), we use $r = \mathbf{R}_1 + j\mathbf{a} = \mathbf{R}_g + k\mathbf{g}$ to get j and k. Then we can easily get the intersect point r.

Based on equation (33), and the common distance equation, we have

$$d^2 = (x - r_x)^2 + (y - r_y)^2 + (z - r_z)^2, \qquad (34)$$

$$d(x, y, z) = R_1(R_x, R_y, R_z) + ta(a_x, a_y, a_z), \qquad (35)$$

where $\mathbf{a}(\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z)$ is the real part of $\hat{\mathbf{X}}_1$. From equations (34) and (35), we obtain two different values of t. We substitute these into equation (35) to determine the two top points $d(x, y, x)$ of the line (screw) segment.

When the two control screws $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ are parallel (as shown in Figure 30), we let the intersection point $r(r_x, r_y, r_x)$ become the point $R(R_x, R_y, R_z)$, which is the origin radius. Note that R and r overlie each other, and their overlapping lines are presented as R_1 and R_2 in Figure 30. From equations (34) and (35), along with this characteristic, d becomes simple to determine:

$$d^2 = (x - R_x)^2 + (y - R_y)^2 + (z - R_z)^2 = t^2(a_x^2 + a_y^2 + a_z^2). \qquad (36)$$

For any ruling, we just need to find the intersection point (central point) of the ruling and its displacement axis along with applying the above method, then the ruling is clipped and shown as a line segment. For example, to show a Bezier surface segment with more than two control screws on the screen, we need to clip all rulings of the last level interpolation and the first and last control screws of the Bezier surface based on their central points using the above method. Figures 31, 32, 33 and 34 show the computer implementation results.

7 Conclusions

This paper explores the direct or indirect applicabilities of the dual de Casteljau algorithm for ruled surfaces with particular shapes. The geometrical characteristics of these shapes are examined and found to be strongly related to the dual de Casteljau algorithm. Accordingly, the variations of the dual de Casteljau algorithm are developed. These variations not only obey the characteristics of the original dual de Casteljau

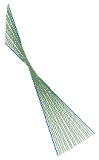


Figure 31. The clipped screws with two non-parallel control screws.

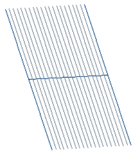


Figure 32. The clipped screws with two parallel control screws.

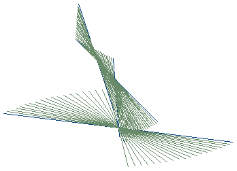


Figure 33. The second level interpolated screws with four control screws.

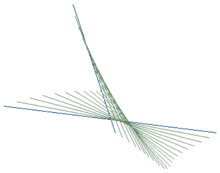


Figure 34. The last level interpolated screws (the ruled surface) with four control screws.

algorithm, but also are suitable for creating particular ruled surfaces. The methods discussed are expected to have direct application, for example, in the areas of movie making, where computer-based generation of successive image frames of motion sequences is required. Also the methods could be used in the planing and production of smooth body trajectories, such as for particular shaped machine parts. The paper has shown the systematic approach of the dual space drawing methods of the particular geometric shapes in CAGD design.

Nevertheless, research in this area is far from complete, and some related topics can be suggested for further study in the future. The dual de Casteljau algorithm and its variations are on the basis of the geometrical representation rather than the functional representation as the real de Casteljau algorithm. Thus, the dual space control screws are more difficult to manipulate than the real space control points. Although Ding [3] and Sprott and Ravani [15] have studied to a certain degree the dual De Boor algorithm, its local control power needs to be further investigated to be applied to and to improve the dual de Casteljau algorithm.

Acknowledgements

The authors acknowledge Douglas Iannucci and Neville Cockburn providing constructive comments for this paper.

References

[1] Carmo, M (1976). Differential geometry of curves and surfaces. Prentice-Hall, Inc. Englewood Cliffs, New Jersey.

[2] Ding, Rena and Yan Zhang(2004). Dual space drawing methods of cones. In proceedings of the International conference on computer graphics and interactive techniques in Australasia and South East Asia. P.p.148-154.

[3] Ding, Rena (2002). Drawing ruled surfaces using the dual De Boor algorithm. Proceedings of Computing: The Australasian Theory Symposium CATS 2002. Elsevier Science B.V.

[4] Farin, G. (1988). Curves and surfaces for computer aided geometric design: A practical guide. Academic Press, Inc., Boston, Mass.

[5] Ge, Q. Jeffrey and Ravani, B.(1994). On representation and interpolation of line-segments for computer aided geometric design. In *ASME Design Automation Conference*, volume 69-1, pages 191-198.

[6] Hill, F.S. Jr. (1990). Computer graphics. Macmillan Publishing Company, New York.

[7] Juttler, B.(1998). Rotation minimizing spherical motions. Advances in Robot Kinematics: Analysis and Control. PP.413-422 Kluwer Academic Publishers.

[8] Leopoldseder, S.(2001). Algorithms on cone spline surfaces and apatial osculating arc splines. Computer Aided Geometric Design 18(2001), 505-530.

[9] McCarthy, J. M. (1990). Introduction to theoretical kinematics. MIT Press, Cambridge, Massachusetts.

[10] Parkin, I.A. (1997). Unifying the geometry of finite displacement screws and orthogonal matrix transformations. Mech. Mach. Theory 32 (8) pp. 975-991.

[11] Piegl, L., and Tiller, W.(1987). Curve and surface constructions using rational G-splines. Computer Aided Design, 19(9):485-98, 1987.

[12] Plűcker, J. (1865). On a new geometry of space. Phil. Trans. Roy. Soc., 155: 725-791.

[13] Pottmann, H., Wallner,J. (2001),Computational Line Geometry, Springer Verlag, Heidelberg, Berlin, ISBN 3-540-42058-4, Pp.565.

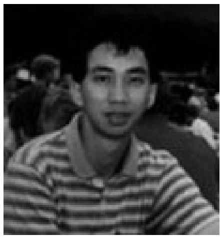
[14] Ravani, B. and Wang, J.W.(1991). Computer aided geometric design of line constructs. *Transactions of the ASME Journal of Mechanical Design*, 113:363-371.

[15] Sprott, K. and Ravani, B. (1997) Ruled surfaces, Lie groups, and mesh generation. 1997 ASME Engineering Techical Conferences, Sep. 14-17, 1997, Sacramento, California,USA.

[16] Srinivasan, Lakshmi N. and Ge, Q. Jeffrey(1997). Fine tuning of rational B-spline motions. Proc. of the ASME Design Engineering Technical Conference. pp.1-10.

[17] Study, E. (1903). Die geometric der dynamen. Leipzig.

Rena Ding received B.E. in Management Engineering from Tongji University and M.S. in Computer Science from the University of Sydney. She is Currently a Ph.D student in Computer Science in University of Western Sydney.



Yan Zhang received B.S. and M.S in Computer Science from Wuhan University and Ph.D in Computer Science from the University of Sydney. He is Currently an associate professor in Computer Science in University of Western Sydney.