

Subdivision Algorithms for Ruled Surfaces

Boris Odehnal

*Institute of Discrete Mathematics and Geometry, Vienna University of Technology
Wiedner Hauptstr 8-10/104, Wien, Austria
email: boris@geometrie.tuwien.ac.at*

Abstract. Recent research has produced results on subdivision in arbitrary manifolds. These results can be applied to the manifold of lines and thus we can create subdivision schemes especially for ruled surfaces. We present different methods for refining discrete models of ruled surfaces: An algorithm combining subdivision and projection to the manifold of lines in Euclidean three-space. A further algorithm combines subdivision for the striction curve with geodesic subdivision in the Euclidean unit sphere. The third method is based on the Denavit-Hartenberg-Method for serial robots. We refine the sequence of motions of the Sannia frame by means of geodesic subdivision in the group of Euclidean motions.

Key Words: Subdivision, ruled surface, striction curve, geodesic subdivision, Euclidean motion, Sannia frame

MSC: 53A25, 65Y25, 52-xx, 68U07

1. Introduction

1.1. Related work

Ruled surfaces appear in many different areas in CAD and CAGD such as in surface approximation [3, 17], wire cutting [3], milling, tool path planning and robot motion planning [18, 19]. Even the recognition and reconstruction of ruled surfaces was studied in [14, 15]. So far almost all of the approaches deal with approximations in terms of parameterized ruled surfaces. There are also some remarkable contributions to the approximation of developable ruled surfaces [13, 22] dealing with different geometric approaches.

In [21] variational subdivision schemes for ruled surfaces and line congruences were introduced. A finite sequence of lines is called a *discrete ruled surface*. It is refined by inserting intermediate lines to the given ones. The presented method uses a triangulation of the discrete model including the estimated lines and leads to the minimization of a quadratic functional. This is a time consuming method although only a system of linear equations (in the coordinates of the endpoints of the intermediate line segments to be inserted) has to be solved.

Subdivision schemes for curves and surfaces were originally developed for data in affine spaces. Nowadays not only the convergence analysis of subdivision schemes is done even schemes on manifolds and on Lie groups are well studied and their possible applications are outlined in [29, 30].

The Klein model of line space is a useful tool for approximation as well as interpolation problems in line space. It also appeared in a slightly modified version for approximation and interpolation problems in [16].

1.2. Motivation

With today's knowledge on subdivision schemes for curves, surfaces, and on manifolds it is possible to define subdivision schemes directly for the set of geometric objects where we want to model in, i.e., the set of lines. Methods for this purpose presented until now consider ruled surfaces as ruled surface strips. The choice of boundary curves enter the construction and thus the approximation. Defining a subdivision scheme in the manifold of lines allows to eliminate the arbitrarily chosen directrices on the ruled surface. This approach is different from the techniques presented so far. We consider ruled surfaces not as ruled surface strips: Lines are unbounded. Further we do not aim at any parameterization or representation like a $(1, n)$ -Bézier surface. We present and describe these methods in order to show how elegant subdivision for arbitrary manifolds can be applied in surface design.

On the other hand there is one special curve called *central curve* or *striction curve* on each ruled surface. Together with the striction curve and the so called *spherical image of the rulings* (which is nothing but a curve in the Euclidean unit sphere) a ruled surface in Euclidean three-space is uniquely defined. So we can combine a subdivision scheme for curves which is applied to an estimated central curve and a *geodesic subdivision* on the Euclidean unit sphere in order to refine the given data of a discrete ruled surface.

A third approach in ruled surface design by means of subdivision uses ideas from kinematics. The discrete ruled surface can be seen as a set of rotation axes of a serial robot. The Denavit-Hartenberg method allows to assign a sequence of Cartesian coordinate systems to the serial robot (or equivalently to its axes). The Euclidean motions transforming each coordinate system in the subsequent one can be seen as points in the group of Euclidean motions. Since subdivision in this group is available we can use this for the design of ruled surfaces too.

1.3. Contributions of the paper

Assume we are given a finite sequence of lines which will henceforth be called *discrete ruled surface* \mathcal{R} . We aim at a refinement of \mathcal{R} , that means, we want to insert new lines in between the old ones. After infinitely many steps of the procedure we want to obtain a smooth surface. For practical reasons C^1 and C^2 transitions are sufficient.

We present three different subdivision techniques: The *projection algorithm* is a combination of projection and subdivision. The second algorithm will be called *combined algorithm* since it combines subdivision in Euclidean space \mathbb{R}^3 applied to a curve and geodesic subdivision on the Euclidean unit sphere S^2 . Finally we use the Denavit-Hartenberg method to refine \mathcal{R} .

The paper is organized as follows: In Section 2 we provide the necessary facts on line geometry, difference geometry, and differential geometry. Section 3 describes the different algorithms and Section 4 shows how the algorithms work. Finally we conclude in Section 5.

Our main goal is to show how these algorithms work. Geometric ideas help to adapt the techniques of curve refinement in order to make them accessible for the design of ruled surfaces.

2. Geometric background

2.1. Line geometry

In the following we work in Euclidean three-space \mathbb{R}^3 . Points P are represented by Cartesian coordinates $\mathbf{p} = (p_1, p_2, p_3)^T$. The space is equipped with the canonical scalar product which reads $\langle \mathbf{u}, \mathbf{v} \rangle = u_1 v_1 + u_2 v_2 + u_3 v_3$ for any pair of vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$. The scalar product defines a cross product $\mathbf{u} \times \mathbf{v}$ for any pair of vectors \mathbf{u} and \mathbf{v} . The cross product vanishes, if \mathbf{u} and \mathbf{v} are linearly dependent. Otherwise $\mathbf{u} \times \mathbf{v}$ is a vector orthogonal to both \mathbf{u} and \mathbf{v} , and its length equals the area of the parallelogram spanned by \mathbf{u} and \mathbf{v} .

A line L in Euclidean space can be represented by a parameterization such as

$$\mathbf{l}(t) = \mathbf{a} + t\mathbf{l} \quad t \in \mathbb{R},$$

where A is a point on L and \mathbf{l} is a unit vector parallel to L . In order to get rid of the point A we use the well known concept of Plücker coordinates, see e.g. [23, 31], and define

$$L = (\mathbf{l}, \bar{\mathbf{l}}), \quad \text{where} \quad \bar{\mathbf{l}} = \mathbf{a} \times \mathbf{l}.$$

Obviously there is no difference which point on L we choose since $\mathbf{a}' = \mathbf{a} + \gamma\mathbf{l}$ (with $\gamma \in \mathbb{R}$) leads to $\mathbf{a}' \times \mathbf{l} = \mathbf{a} \times \mathbf{l}$. The vector $\bar{\mathbf{l}}$ is called *momentum vector* of L . The vectors \mathbf{l} and $\bar{\mathbf{l}}$ are mutually orthogonal and together with the normalization condition for the vector \mathbf{l} the Plücker coordinates $(\mathbf{l}, \bar{\mathbf{l}})$ of L satisfy

$$M^4 : \langle \mathbf{l}, \mathbf{l} \rangle = 1, \quad \langle \mathbf{l}, \bar{\mathbf{l}} \rangle = 0. \quad (1)$$

Note that lines in Euclidean three-space are oriented according to the orientation of \mathbf{l} .

The manifold $M^4 \subset \mathbb{R}^6$ given by Eq. (1) is a point model for the set of oriented lines in Euclidean three-space. To each oriented line in \mathbb{R}^3 we find exactly one point on the manifold M^4 . Conversely, we find exactly one oriented line in \mathbb{R}^3 to each point on M^4 . Smooth curves (polygons) contained in M^4 correspond to smooth (discrete) one-parameter families of lines, so called *ruled surfaces*.

2.2. Projection onto M^4

Assume we are given a polygon in the model space \mathbb{R}^6 . The vertices of the polygon correspond to lines in \mathbb{R}^3 , if they are contained in M^4 . Now we apply a subdivision scheme to the polygon and end up with a finer one. The new vertices of the finer polygon are in general not contained in the manifold M^4 . We replace these vertices by the closest points in M^4 . For that we have to define what does *closest* mean.

Let $P = (\mathbf{p}, \bar{\mathbf{p}}) \in \mathbb{R}^6 \setminus M^4$ be a point in the model space but not on M^4 . We define an objective function F to be minimized as

$$F(\mathbf{x}, \bar{\mathbf{x}}) = \langle \mathbf{x} - \mathbf{p}, \mathbf{x} - \mathbf{p} \rangle + \langle \bar{\mathbf{x}} - \bar{\mathbf{p}}, \bar{\mathbf{x}} - \bar{\mathbf{p}} \rangle. \quad (2)$$

A solution $X = (\mathbf{x}, \bar{\mathbf{x}}) \in \mathbb{R}^6$ has to fulfill Eqs. (1). So we define the Lagrange function

$$L(\mathbf{x}, \bar{\mathbf{x}}, \lambda_1, \lambda_2) = F(\mathbf{x}, \bar{\mathbf{x}}) - \lambda_1(\langle \mathbf{x}, \mathbf{x} \rangle - 1) - 2\lambda_2 \langle \mathbf{x}, \bar{\mathbf{x}} \rangle. \quad (3)$$

The necessary condition for $(\mathbf{x}, \bar{\mathbf{x}}, \lambda_1, \lambda_2)$ to be a local minimizer of L from (3) is that it solves the system of linear equations comprising the partial derivatives of L with respect to all the

unknowns \mathbf{x} , $\bar{\mathbf{x}}$, λ_1 , and λ_2 . So we have

$$\begin{aligned} \mathbf{x} - \mathbf{p} + \lambda_1 \mathbf{x} + \lambda_2 \bar{\mathbf{x}} &= 0, \\ \bar{\mathbf{x}} - \bar{\mathbf{p}} + \lambda_2 \mathbf{x} &= 0, \\ 1 - \langle \mathbf{x}, \mathbf{x} \rangle &= 0, \\ \langle \mathbf{x}, \bar{\mathbf{x}} \rangle &= 0. \end{aligned}$$

Solving this system of equations results in solving an algebraic equation of degree four. In order to avoid lengthy computations and ambiguous solutions we settle for a weaker solution by simplifying the projection.

Given any $P = (\mathbf{p}, \bar{\mathbf{p}}) \in \mathbb{R}^6$ we can compute the Plücker coordinates $(\mathbf{l}, \bar{\mathbf{l}})$ of a line L close to P in the following way: We let $\mathbf{l} = \mathbf{p}/\|\mathbf{p}\|$ and then we apply the Gram-Schmidt procedure to the pair of vectors $(\mathbf{l}, \bar{\mathbf{p}}/\|\mathbf{p}\|)$ and find

$$(\mathbf{l}, \bar{\mathbf{l}}) = \left(\frac{\mathbf{p}}{\|\mathbf{p}\|}, \frac{\bar{\mathbf{p}}}{\|\mathbf{p}\|} - \frac{\mathbf{p}\langle \mathbf{p}, \bar{\mathbf{p}} \rangle}{\|\mathbf{p}\|^3} \right). \quad (4)$$

This pair of vectors satisfies the conditions (1) and thus they are coordinates of a line in \mathbb{R}^3 . A geometric interpretation of this Gram-Schmidt procedure for Plücker coordinates is illustrated in Fig. 1. The vector \mathbf{l} aims at some point of the Euclidean unit sphere S^2 and is obtained by normalizing \mathbf{p} . The momentum vector $\bar{\mathbf{l}}$ of the closest line needs to be perpendicular to \mathbf{l} and is thus obtained by orthogonal projection to \mathbf{l} 's normal plane $\mathbf{l}^\perp = \mathbf{p}^\perp$, which is illustrated in Fig. 1.

There are further possibilities for the projection of points in \mathbb{R}^6 to the manifold M^4 which should be mentioned here:

It is well known, that the points which are not contained in M^4 represent regular linear line complexes, see [23, 31]. A linear line complex comprises the set of path normals of a

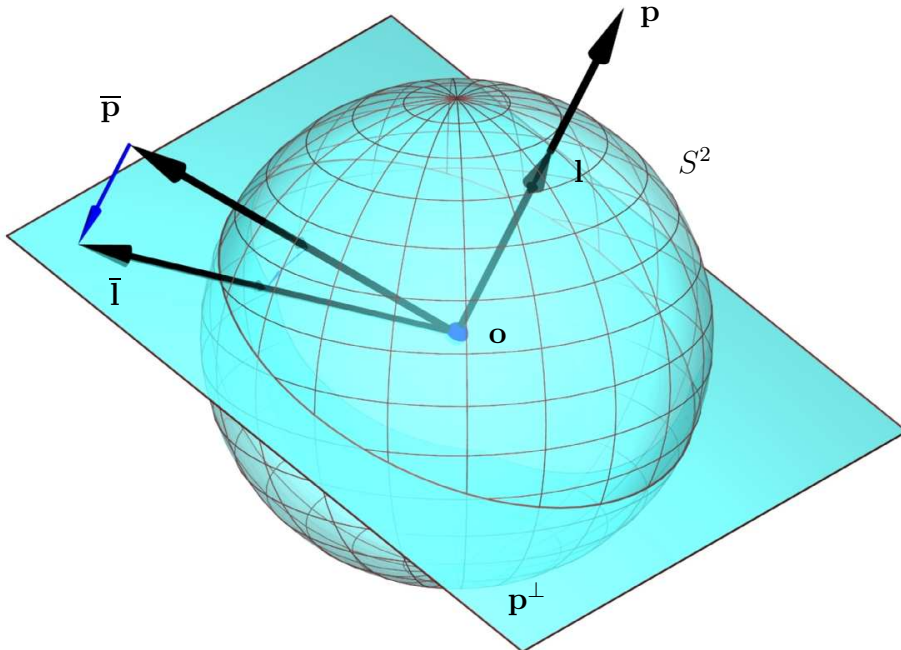


Figure 1: Computing the closest lines

certain well defined helical motion. Thus a point $(\mathbf{p}, \bar{\mathbf{p}})$ (with $\langle \mathbf{p}, \bar{\mathbf{p}} \rangle \neq 0$) can be projected to M^4 by simply computing the axis $A = (\mathbf{a}, \bar{\mathbf{a}})$ of the complex, which is the axis of the corresponding helical motion. For the axis A we have $(\mathbf{a}, \bar{\mathbf{a}}) = (\mathbf{p}, \bar{\mathbf{p}} - \langle \mathbf{p}, \bar{\mathbf{p}} \rangle / \langle \mathbf{p}, \mathbf{p} \rangle \mathbf{p})$ (see [23, 31]), which obviously agrees with Eq. (4) up to the factor $\|\mathbf{p}\|$. (We do not have to be afraid of $\|\mathbf{p}\| = 0$, because later, when we apply averaging procedures to proper lines we always can assume $\mathbf{p} \neq 0$ and orientations will not change suddenly.)

Finally we remark that the formula given in (4) can be found in a different way: Oriented lines in Euclidean three-space can be described by dual unit vectors $L = \mathbf{l} + \varepsilon \bar{\mathbf{l}}$, $\|\mathbf{l}\| = 1$, $\langle \mathbf{l}, \bar{\mathbf{l}} \rangle = 0$, $\varepsilon^2 = 0$ (see [27, 31]). Affine combinations of dual unit vectors do in general not produce dual unit vectors. Thus we could project them to the unit sphere in the dually extended Euclidean three-space in order to find the line closest to this dual vector. The dual vectors represent screws (see [1, 9, 27, 31]). The computation of the screw axis (i.e., the axis of a helical motion) results in scaling a vector with a dual number and leads to the formula given in (4).

2.3. Ruled surfaces

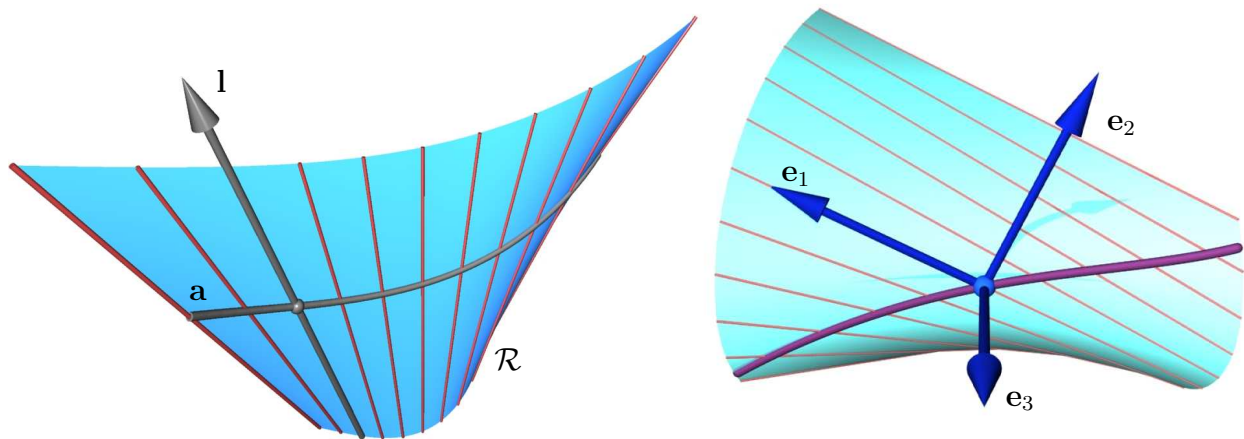


Figure 2: Left: Ruled surface with a directrix and some generators, Right: Sannia frame

A ruled surface \mathcal{R} is a (smooth or discrete) one-parameter family of lines. We assume that $\mathbf{a}(u): I \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is a parameterization of a curve a in Euclidean three-space \mathbb{R}^3 and $\mathbf{l}(u): I \rightarrow S^2$ is a parameterization of a curve l in the Euclidean unit sphere S^2 . The ruled surface \mathcal{R} with *directrix* a and *spherical image* l is parameterized by

$$\mathcal{R} = \mathbf{a} + v\mathbf{l} \quad (5)$$

over $I \times \mathbb{R} \subset \mathbb{R}^2$. Curves on \mathcal{R} defined by fixed u are called *generators* or *rulings*. A part of a ruled surface can be seen in Fig. 2. Any curve $\mathbf{a} + v(u) \cdot \mathbf{l}$ on \mathcal{R} , which intersects all of the rulings in exactly one point, can serve as a directrix. In the following we want to replace \mathbf{a} by a special curve.

At any regular surface point the vectors $\mathcal{R}_{,u} = \dot{\mathbf{a}} + v\dot{\mathbf{l}}$ and $\mathcal{R}_{,v} = \mathbf{l}$ are linearly independent. Here $\dot{}$ indicates differentiation with respect to the only parameter u in both \mathbf{a} and \mathbf{l} , respectively. The normal vectors of \mathcal{R} at the points on one ruling \mathcal{R}_0 are thus given by

$$\mathbf{n} = \dot{\mathbf{a}} \times \mathbf{l} + v\dot{\mathbf{l}} \times \mathbf{l} \quad (6)$$

and obviously, they depend linearly on v . Therefore there is a projective mapping assigning to each point $P_0 \in \mathcal{R}_0$ its tangent plane $T_{P_0}\mathcal{R}$.

The tangent plane T_ω at \mathcal{R}_0 's ideal point contains the ideal points $(0, \mathbf{l})$ and $(0, \dot{\mathbf{l}})$ – represented by their homogeneous Cartesian coordinates – which span the tangent to \mathcal{R} 's ideal curve at the ideal point $(0, \mathbf{l})$. There exists a tangent plane T_S of \mathcal{R} which is orthogonal to T_ω . We call T_S the *central plane* and the uniquely determined point S of contact *striction point* or *central point*. The central points form the *central curve* or *striction curve* on \mathcal{R} . It is easy to verify that the striction curve can be parameterized by

$$\mathbf{s} = \mathbf{a} - \frac{\langle \dot{\mathbf{a}}, \dot{\mathbf{l}} \rangle}{\langle \dot{\mathbf{l}}, \dot{\mathbf{l}} \rangle} \mathbf{l}. \quad (7)$$

Rulings with indetermined denominator of (7) are called *cylindrical rulings*. Ruled surfaces carrying only cylindrical rulings are called *cylinders*.

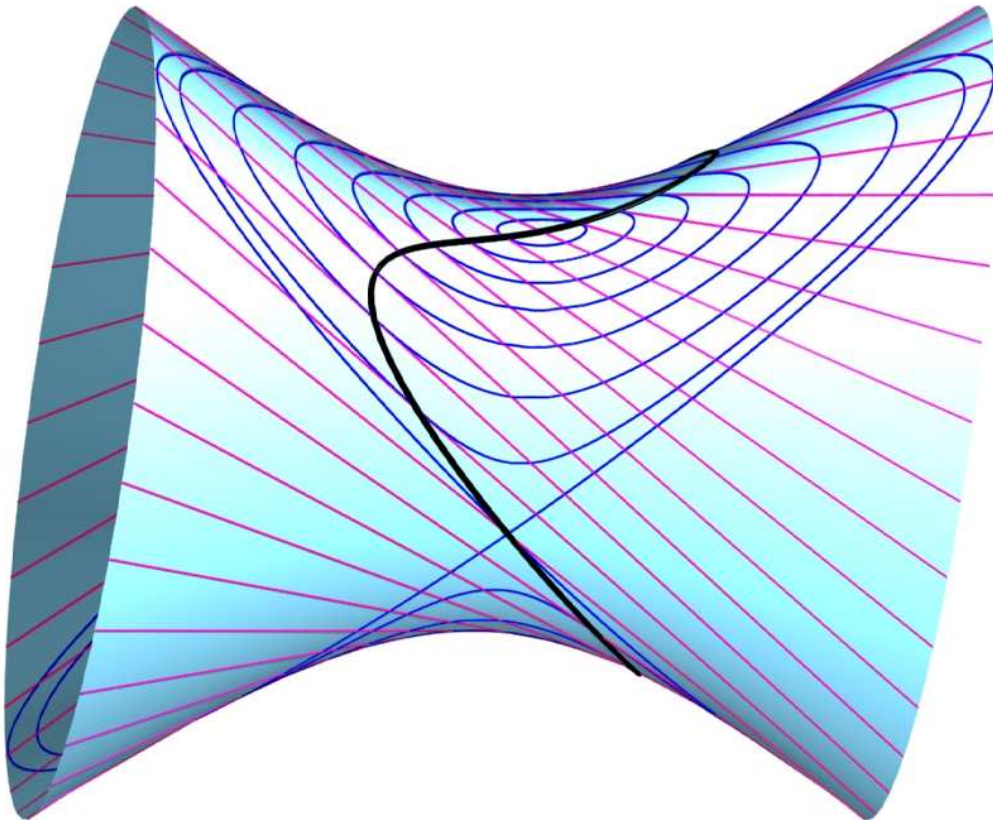


Figure 3: Striction curve as locus of points on the rulings where the Gaussian curvature attains its maximum

The striction curve is the special directrix we have been looking for. The Gaussian curvature considered to be a function on the ruling reaches its maximum exactly at the striction point. This can easily be seen from Lamarle's formula for the Gaussian curvature of a ruled surface [23]. Consequently the curves of constant Gaussian curvature on \mathcal{R} touch the rulings exactly at the respective striction points which is illustrated in Fig. 3.

In the following we assume that ruled surfaces are given by their striction curve and their

spherical image. Before we try to find the striction curve in a different way, we recall some other differential geometric properties of ruled surfaces.

We call a generator \mathcal{R}_0 *torsal*, if the tangent planes do not wind about \mathcal{R}_0 . A ruled surface consisting only of torsal generators is called *torsal ruled surface*. These surfaces can be mapped isometrically to the Euclidean plane (at least locally) and are therefore called *developable ruled surfaces*. (They consist of parts of cylinders, cones, planes, and the surfaces traced by the tangents of spatial curves, provided that they are at least two times differentiable.)

For torsal generators the vectors $\dot{\mathbf{a}} \times \mathbf{l}$ and $\dot{\mathbf{i}} \times \mathbf{l}$ appearing in (6) are linearly dependent and thus a torsal generator is characterized by

$$\det(\dot{\mathbf{a}}, \dot{\mathbf{i}}, \mathbf{l}) = 0.$$

Later when we approximate ruled surfaces with subdivision schemes in the group of Euclidean motions, we need to have a geometrically favorable coordinate system attached to the ruled surface. Therefore we define a right handed basis which in classical literature frequently appears as Sannia *frame*. An example is shown in Fig. 2. The moving frame is centered at the central point S and its vectors are

$$\mathbf{e}_1 = \mathbf{l}, \quad \mathbf{e}_2 = \dot{\mathbf{i}} / \|\dot{\mathbf{i}}\|, \quad \mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2. \quad (8)$$

The lines $\mathbf{s} + [\mathbf{e}_2]$ and $\mathbf{s} + [\mathbf{e}_3]$ are called *central normal* and *central tangent*, respectively.

2.4. Difference geometric approach

The striction curve and the Sannia frame can be found in a different way. Assume that the ruled surface \mathcal{R} is analytic, i.e., any directrix \mathbf{l} and the spherical image \mathbf{r} can be expanded in Taylor series.

Assume now that \mathcal{R}_0 is the generator at a certain parameter value u_0 and \mathcal{R}_ε is the generator corresponding to the value $u_0 + \varepsilon$ with sufficiently small ε . Then we compute the common normal $N_{0,\varepsilon}$ of \mathcal{R}_0 and \mathcal{R}_ε and the respective intersection points F_0 and F_ε . We observe that N_0 converges to the striction point S_0 if \mathcal{R}_ε moves to \mathcal{R}_0 while $\varepsilon \rightarrow 0$.

For the difference geometric treatment of ruled surfaces see the monograph [25]. A good approximation of the striction curve by a polygon is given in [21]: On each ruling \mathcal{R}_0 we compute the midpoint S_0 of the two pedal points $F_{-\varepsilon,0}$ and $F_{0,\varepsilon}$. This is illustrated in Fig. 5. The approximation of the striction curve by the polygon of midpoints has even quadratic convergence as shown in [21].

Problems with the method described so far may occur when two consecutive rulings R_i and R_{i+1} of the discrete ruled surface are parallel. In that case the common normal is not unique and we proceed in the way shown in Fig. 4: We compute the discrete version of the striction curve from the left and the right hand side of \mathcal{R}_i and \mathcal{R}_{i+1} , respectively. Then we project the left most (F_i^+) and right most (F_{i+1}^-) pedal point orthogonally to \mathcal{R}_{i+1} and \mathcal{R}_i , respectively. We obtain $F_{i+1}^{-\prime} \in \mathbb{R}_i$ and $F_i^{+\prime} \in \mathbb{R}_{i+1}$ and the common normal of \mathcal{R}_i and \mathcal{R}_{i+1} shall connect $M_i = (F_i^+ + F_{i+1}^{-\prime})/2$ and $M_{i+1} = (F_{i+1}^- + F_i^{+\prime})/2$. This is also done for more than two parallel consecutive rulings. Now the points M_i and M_{i+1} serve as pedal points of a common normal of two consecutive rulings and again we are able to compute the striction points as midpoints in the way shown above.

In case of intersecting consecutive rulings one has to decide whether the ruled surface should be interpolated or approximated. In the first case the intersection point is a good choice for the central point on both rulings \mathcal{R}_i and \mathcal{R}_{i+1} . In the latter case an average of

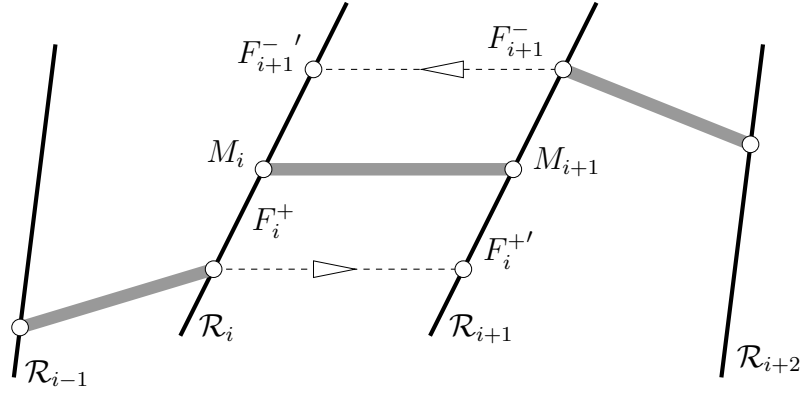


Figure 4: Estimating central points at torsal or cylindrical rulings

the intersection point and the previously computed pedal points would be sufficient, since approximating subdivision schemes result in curves that make sharp corners round. So is the case for subdivision schemes for ruled surfaces.

It is well known and obvious that the common normal of \mathcal{R}_0 and \mathcal{R}_ε converges to the central normal. The pitch of the helical motion transforming \mathcal{R}_ε to \mathcal{R}_0 converges towards the so called *distribution parameter* δ of \mathcal{R} at \mathcal{R}_0 . It is a measure for the winding of the tangent planes of \mathcal{R} along \mathcal{R}_0 and is zero for torsal generators, see [8, 23]. These results can easily be obtained from the view point of kinematics based on dual vector calculus, see [27].

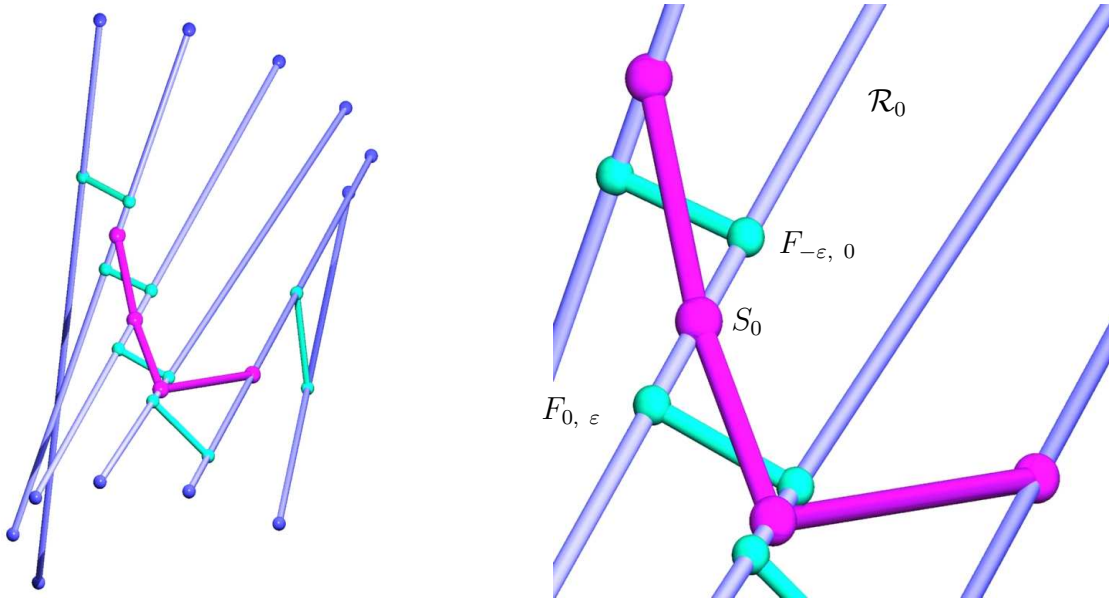


Figure 5: Discrete version of the striction curve

2.5. Interpolatory subdivision for curves

We are interested in the interpolation of given discrete ruled surfaces by finer models. Since we know that discrete / smooth ruled surfaces are represented by polygons / curves in the Klein model, we study subdivision schemes for curves. One famous scheme is the four-point-scheme by DYN et al. [4]. In the following we use the abbreviation DLG-scheme.

Given a finite sequence $(P_i^{\text{old}})_{i \in I}$ of points with coordinate vectors $\mathbf{p}_i^{\text{old}}$, new points are computed according to

$$\begin{aligned} \mathbf{p}_{2i}^{\text{new}} &= \mathbf{p}_i^{\text{old}}, \\ \mathbf{p}_{2i+1}^{\text{new}} &= \left(\frac{1}{2} + w\right)(\mathbf{p}_i^{\text{old}} + \mathbf{p}_{i+1}^{\text{old}}) - w(\mathbf{p}_{i-1}^{\text{old}} + \mathbf{p}_{i+2}^{\text{old}}), \end{aligned} \quad (9)$$

where w is a shape parameter. The DLG-scheme is known to converge towards a C^1 -curve in the limit, if $0 \leq w < \frac{1}{8}$. Small w are straightening the curve towards the initial polygon. If the parameter w is chosen larger than $1/8$ the scheme produces fractal like curves.

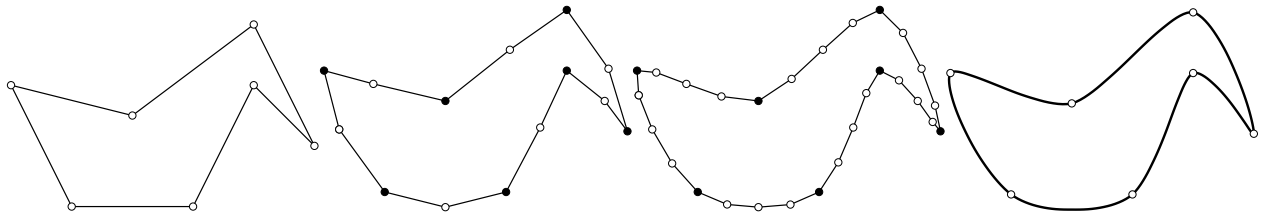


Figure 6: Some steps of the four-point-scheme and limit curve ($w = 1/16$)

The DLG-scheme is a special case of an interpolatory scheme. Subdivision schemes with arbitrarily high smoothness in the limit can be generated [24]. Ternary schemes insert two new points in between two old points in each step and can be used to produce C^2 -curves in the limit, see [7].

2.6. Approximating schemes for curves

Sometimes it is sufficient to perform a refining technique to geometric objects that produces only an approximating family of these objects. For curves there exists the well known scheme by CHAIKIN [2]. It refines a polygon by simply cutting corners away. The *corner cutting* can be applied to polygons according to different masks – that is the matrix of coefficients of the affine combinations defining the scheme – such that it produces splines consisting of curves which are joined with highest possible continuity in the limit, see [24].

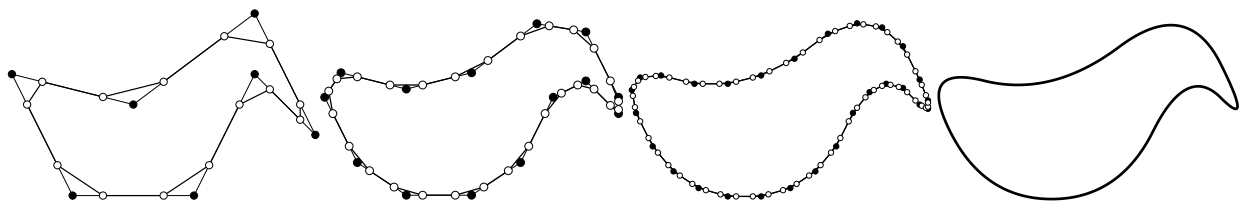


Figure 7: Some steps of Chaikin's algorithm and limit curve

A simple and well known kind of approximating scheme produces a sequence of parabolae joined with C^1 -continuity at the midpoints of the initial polygon. New points are computed from the old ones by

$$\begin{aligned} \mathbf{p}_{2i}^{\text{new}} &= \frac{3}{4} \mathbf{p}_i^{\text{old}} + \frac{1}{4} \mathbf{p}_{i+1}^{\text{old}}, \\ \mathbf{p}_{2i+1}^{\text{new}} &= \frac{1}{4} \mathbf{p}_i^{\text{old}} + \frac{3}{4} \mathbf{p}_{i+1}^{\text{old}}. \end{aligned} \quad (10)$$

For open curves the first and the last point are left fixed. The second point of the new ones is the midpoint of $\mathbf{p}_1^{\text{old}}$ and $\mathbf{p}_2^{\text{old}}$. Analogously we proceed with the last segment of the input

polygon. Applying higher order schemes we have to take care of a certain number of input points at the beginning and at the end of the list depending on the order of the scheme.

The coefficients in the mask can be easily obtained from Pascal's triangle [24]. We will make use of this fact when refining data in order to obtain smoother results.

2.7. Geodesic subdivision on the Euclidean unit sphere

The second algorithm for the design of ruled surfaces combines subdivision for the discrete version of the striction curve with a subdivision scheme applied to the direction vectors of the rulings. It is no restriction to assume that all direction vectors are normalized, i.e., they are of unit length. (We have already done so while collecting differential geometric properties of ruled surfaces.)

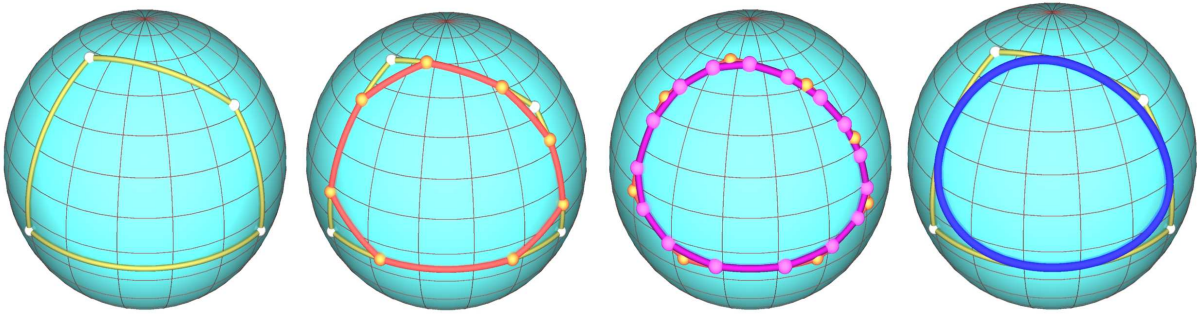


Figure 8: Spherical version of Chaikin's algorithm (left to right): Initial spherical control polygon, first and second step of the spherical corner cutting, and limit curve

Therefore we study a subdivision scheme that works on the unit sphere. In order to show how it works we define the spherical version of Chaikin's algorithm as a geodesic subdivision scheme. This is illustrated in Fig. 8.

As outlined in Sec. 2.6 Chaikin's algorithm means simply cutting away corners of the control polygon. The edges of the control polygon for the planar version of Chaikin's algorithm are carried by straight lines, i.e., the geodesics in the plane.

A spherical control polygon now consists of points in the Euclidean unit sphere, i.e., a sequence of unit vectors. The edges are segments of spherical geodesics (great circles) on the sphere. During the algorithm, midpoints have to be computed. The spherical midpoint $\mathbf{m}(\mathbf{p}_i, \mathbf{p}_{i+1})$ of two points \mathbf{p}_i and \mathbf{p}_{i+1} (both contained in the unit sphere) is now given by

$$\mathbf{m}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \frac{\mathbf{p}_i + \mathbf{p}_{i+1}}{\|\mathbf{p}_i + \mathbf{p}_{i+1}\|}, \quad (11)$$

where $\|\mathbf{v}\|$ is the Euclidean length of \mathbf{v} . The orientation of rulings will not change in a small neighborhood of a regular ruling, so we can assume $\mathbf{p}_i \neq \mathbf{p}_{i+1}$ and thus (11) is well defined. Eq. (11) allows to apply the Chaikin scheme as defined in Eq. (10) to a spherical polygon. Since

$$(3\mathbf{p}_i + \mathbf{p}_{i+1})/4 = (\mathbf{p}_i + (\mathbf{p}_i + \mathbf{p}_{i+1})/2)/2$$

we have to compute three times a spherical midpoint according to formula (11) in each step and for each segment in order to perform the *geodesic version of Chaikin's algorithm on the Euclidean unit sphere*. Higher order corner cutting schemes can also be defined by

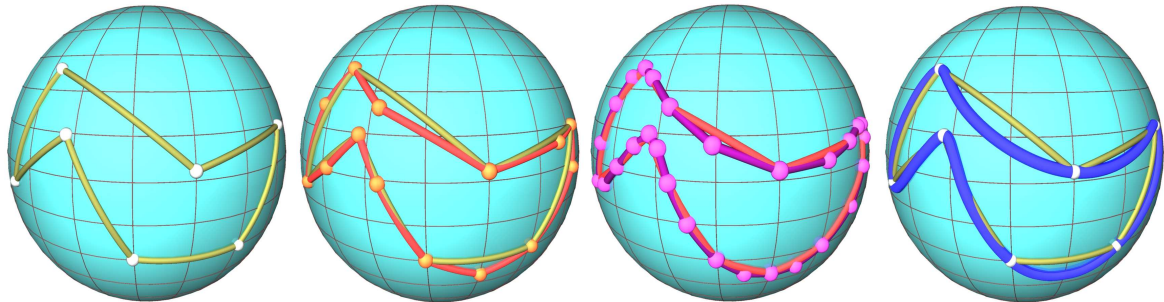


Figure 9: The DLG-scheme on the sphere

iterated computation of midpoints. Thus the spherical and indeed any geodesic version can be computed in the same way.

Affine combinations on the sphere can also be computed via *spherical linear interpolation*. Let \mathbf{p}_i and \mathbf{p}_{i+1} be two unit vectors, i.e., points in the unit sphere, and let further φ be the angle enclosed by them. Then

$$\text{slerp}(\mathbf{p}_i, \mathbf{p}_{i+1}, t) = \frac{\sin(1-t)\varphi}{\sin \varphi} \mathbf{p}_i + \frac{\sin t\varphi}{\sin \varphi} \mathbf{p}_{i+1}, \quad (12)$$

where $0 \leq t \leq 1$ parameterizes the circular arc between \mathbf{p}_i and \mathbf{p}_{i+1} with constant velocity φ . It was introduced into the computer graphics community in [26]. Note that this formula is independent of the dimension of the underlying space. With Eq. (12) each averaging procedures on the sphere can be performed.

In order to adapt the DLG-scheme for the sphere, we manipulate the second equation of (9) and find

$$\begin{aligned} \left(\frac{1}{2} + w\right) (\mathbf{p}_i + \mathbf{p}_{i+1}) - w(\mathbf{p}_{i-1} + \mathbf{p}_{i+2}) &= \frac{1}{2} \mathbf{q}_i + \frac{1}{2} \mathbf{q}_{i+1}, \quad \text{with} \\ \mathbf{q}_i &= -2w\mathbf{p}_{i-1} + (1 + 2w)\mathbf{p}_i \quad \text{and} \quad \mathbf{q}_{i+1} = (1 + 2w)\mathbf{p}_{i+1} - 2w\mathbf{p}_{i+2}. \end{aligned}$$

Consequently the newly inserted point can be constructed by applying linear combinations to three pairs of points. Therefore and with (12) the DLG-scheme can also be applied to data in the sphere. An example is shown in Fig. 9.

2.8. Geodesic subdivision in the group of Euclidean motions

Now we assume that we are given two orthonormal frames

$$\Sigma_i = (\mathbf{s}_i; \mathbf{e}_{1,i}, \mathbf{e}_{2,i}, \mathbf{e}_{3,i}), \quad \Sigma_{i+1} = (\mathbf{s}_{i+1}; \mathbf{e}_{1,i+1}, \mathbf{e}_{2,i+1}, \mathbf{e}_{3,i+1}).$$

There is a uniquely defined *shortest* uniform Euclidean motion (this is either a helical motion, or a rotation, or a translation)

$$\mu: \Sigma_i \rightarrow \Sigma_{i+1}.$$

By shortest (without introducing a metric) we mean that the angle of the rotational part is the smallest possible. The Euclidean motion μ is the geodesic within the group of Euclidean motions joining the two points representing the respective orthonormal frames Σ_i and Σ_{i+1} . We can parameterize this motion with constant speed such that $\mu(0) = \Sigma_i$ and $\mu(1) = \Sigma_{i+1}$. The geodesic midpoint $\Sigma_i^m = (\mathbf{s}_i^m; \mathbf{e}_{1,i}^m, \mathbf{e}_{2,i}^m, \mathbf{e}_{3,i}^m)$ is then given by $\mu(\frac{1}{2}) = \Sigma_i^m$.

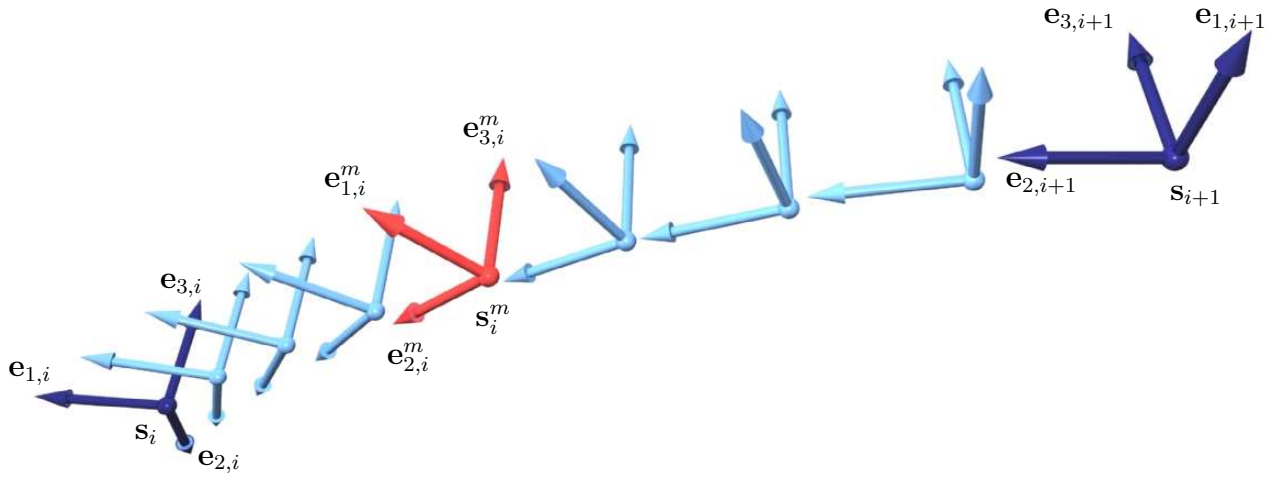


Figure 10: Computing the geodesic midpoint in the group of Euclidean motions

An example is shown in Fig. 10. These facts enable us to modify Chaikin's algorithm – indeed any corner cutting algorithm – and also the DLG scheme for subdivision in the Euclidean motion group, see for example Fig. 11.

There are several types of interpolation and modeling techniques for motion design and motion planning in Study's quadric, see [5, 6, 10, 11, 12].

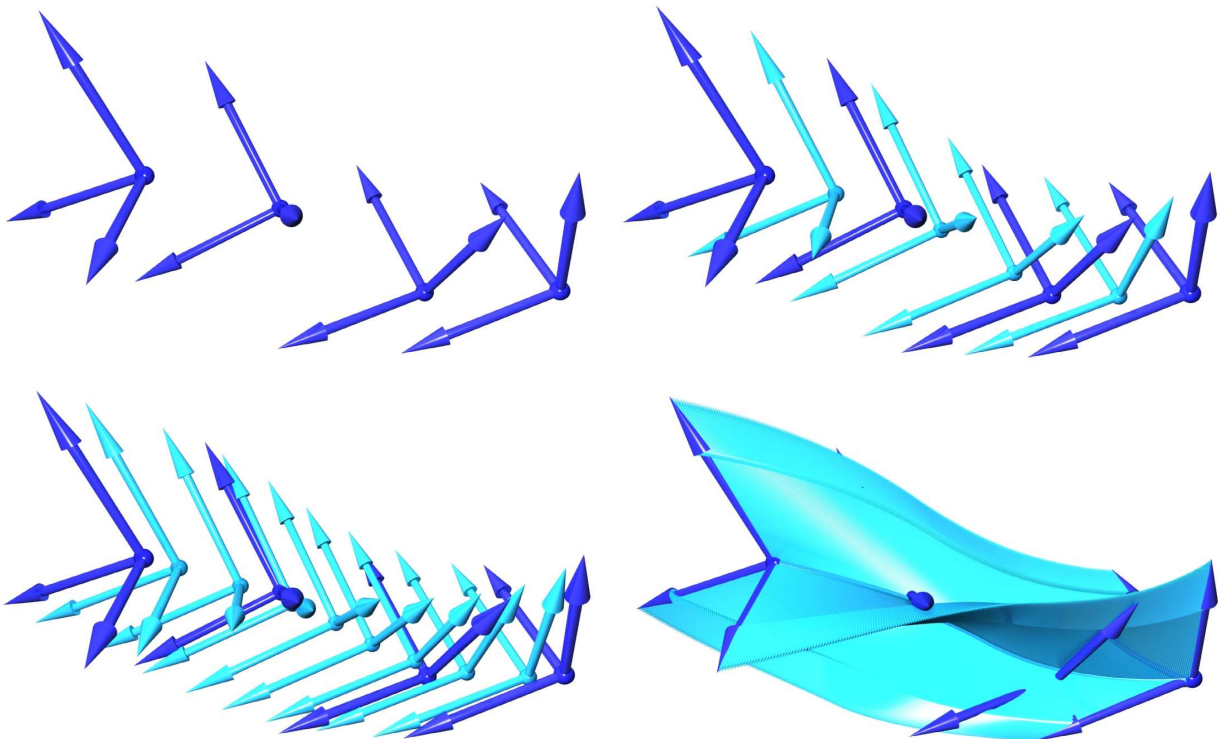


Figure 11: Chaikin's algorithm in the group of Euclidean motions: Initial sequence of orthonormal frames, two steps, and the result after eight steps of geodesic subdivision in the group of Euclidean motions

3. Algorithms

3.1. Projection algorithm

The first algorithm will be called the *projection algorithm* because it uses a projection to the manifold M^4 as an intermediate step.

First we are given an initial list of data lines L_i . These are represented by a list of data points in the Klein model of Euclidean line space. Then we refine the polygon in \mathbb{R}^6 by applying an interpolatory or approximating subdivision scheme. The resulting finer polygon may consist of two types of points: Points which lie on the manifold M^4 and points which do not. Points of the second type do not represent lines in Euclidean three-space \mathbb{R}^3 . Thus we use Eq. (4) in order to find the *nearest* points M^4 .

Repeating the averaging and the projection we obtain in the limit a smooth ruled surface. The necessary convergence results are given in [28, 29, 30].

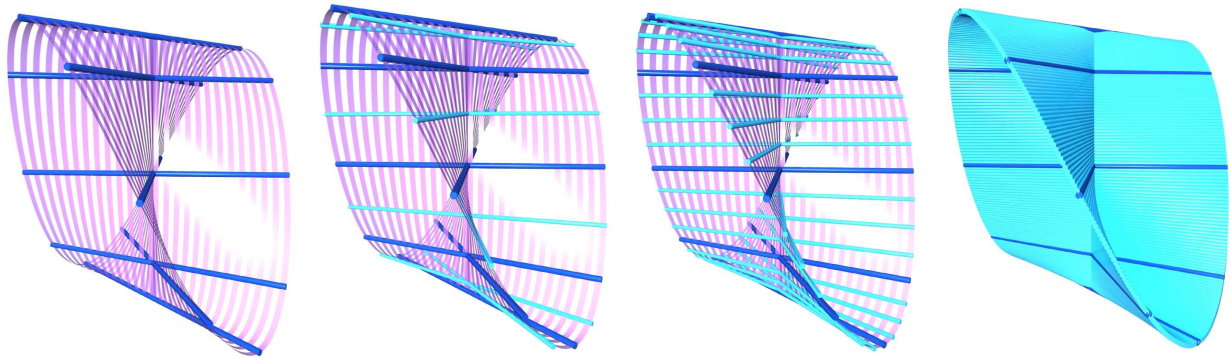


Figure 12: Example 1: The projection algorithm reproduces the straight striction curve of the cylindroid

3.2. Combined algorithm

The second approach can be used to refine a discrete ruled surface in a neighbourhood of the striction curve. Assume we are given an initial list of lines L_i . At first we compute the discrete analogue of the striction curve as shown in Sec. 2.4. Then we compute the spherical image of the discrete ruled surface which results in a sequence of points in the Euclidean unit sphere S^2 . Now we apply a subdivision scheme to the discrete striction curve and a geodesic subdivision scheme in S^2 to the spherical image as shown in Sec. 2.7. Since this algorithm combines ordinary subdivision with geodesic subdivision in the sphere we call this technique *combined algorithm*.

3.3. Denavit-Hartenberg-algorithm

A sequence $(L_i)_{i \in I}$ of lines can be seen as a discrete ruled surface or as the set of axes of a serial robot with $\#I$ revolute joints. The direct kinematics of this serial robot uses the so called Denavit-Hartenberg method, see [1, 9]. To each system a Cartesian coordinate system is attached such that the first axis coincides with the rotation axis between two adjacent systems. The second axis coincides with the common normal of the two adjacent rotation axes. We choose the origin on L_i at the pedal point of the common normal between L_i and L_{i+1} . There are no problems with closed kinematical chains. For open chains (i.e., open ruled surfaces)

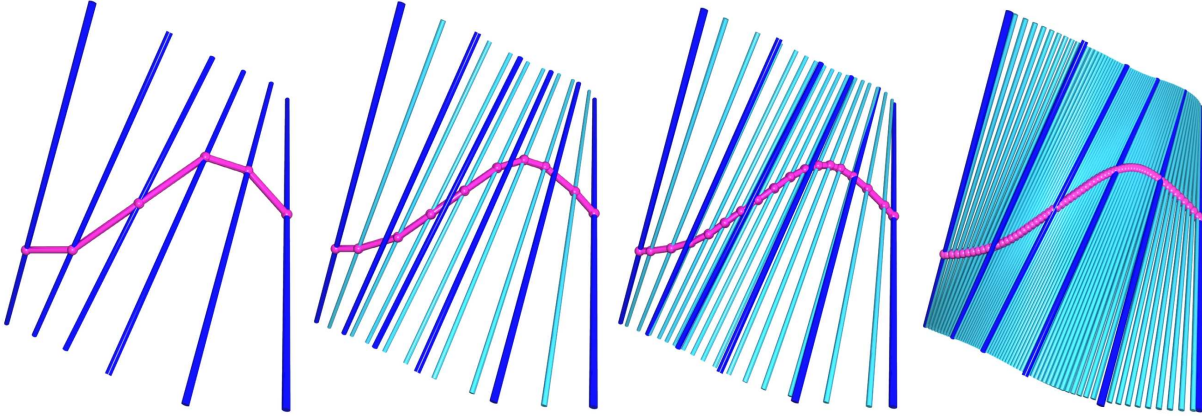


Figure 13: Example 3: Combining Chaikin's rule for the striction curve as well as the spherical Chaikin algorithm for the spherical image

we choose the pedal point on L_1 of L_1 and L_2 , and in an analogous way at the end of the chain.

Each of the Cartesian coordinate systems Σ_i attached to the discrete ruled surface can be seen as an analogue to the Sannia frame as described in Sec. 2.3. There is a uniquely determined shortest Euclidean motion (helical motion rotation, or translation) transforming Σ_i into Σ_{i+1} . By means of geodesic subdivision within the group Euclidean motions (explained in Sec. 2.8) we can refine the given sequence of motions or equivalently the sequence of given Sannia frames.

This algorithm delivers more than rulings of a finer model of the ruled surface. The trace of origins serves as an approximant of the striction curve, and the one-parameter motion of the Sannia frame is also refined.

4. Examples

4.1. Projection algorithm: example 1 & 2

The cylindroid rendered data for the first test of the projection algorithm. Eight lines evenly distributed on the entire surface were picked. Five steps of the algorithm which combined subdivision for the points being the lines images in the Klein model and subsequent projection to the manifold of lines were performed. This example shows that the straight striction curve which is part of a directing line contained in the cylindroid are reproduced by the algorithm.

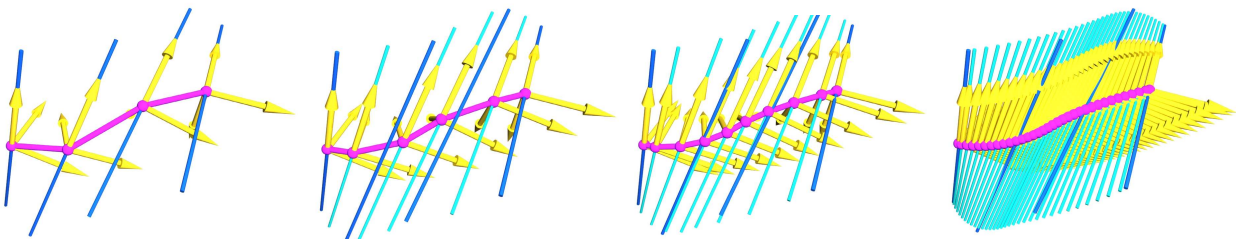


Figure 14: Example 5: The Denavit-Hartenberg-Method shows a way to refine ruled surfaces by means of geodesic subdivision in the group of Euclidean motions

The results can be seen in Fig. 12.

The projection algorithm does not fail even if we have to refine data that comes from a developable ruled surface. The pictures shown in Fig. 15 demonstrate that even the curve of regression which was computed from the set of lines of the finer discrete ruled surface can be recovered. The data in this special example comes from a helical developable.

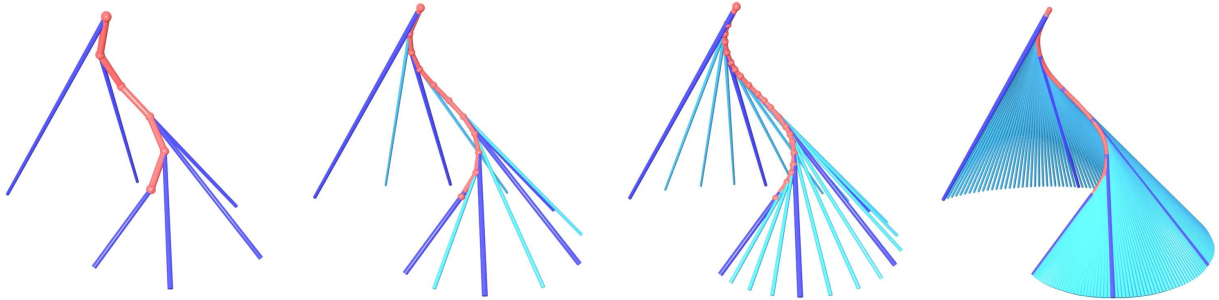


Figure 15: Example 2: Reproducing a helical developable with the projection algorithm

4.2. Combined algorithm: example 3 & 4

Four arbitrarily given lines in Euclidean three-space served as initial sequence of lines (see Fig. 13) for a test of the combined algorithm. We applied one, two, and four rounds of the combined algorithm and obtained the lines of the interpolating finer discrete ruled surface also shown in Fig. 13.

We have tested our algorithm on data from a ruled surface which is of the topological type of a Moebius strip. In Fig. 16 we see the result of the combined algorithm applied to data from the Moebius-like surface.

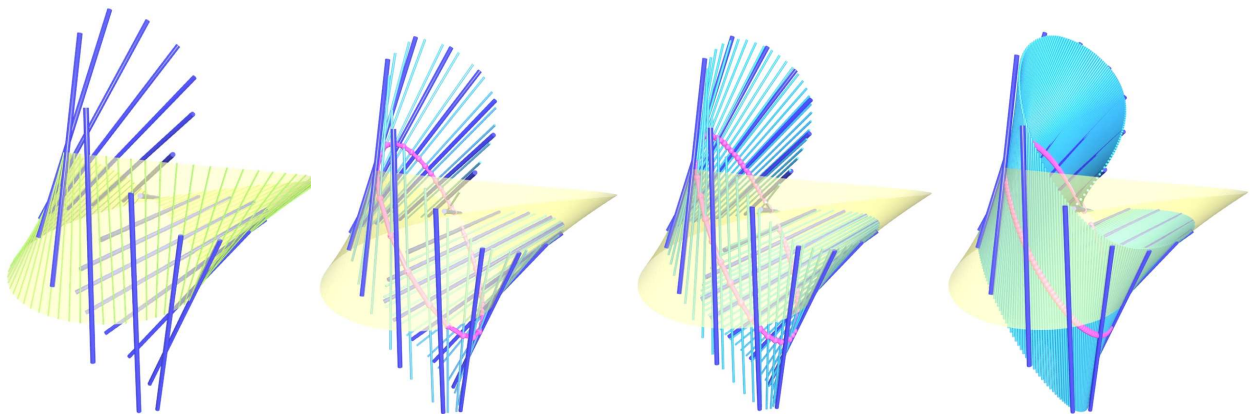


Figure 16: Example 4: Moebius strip approximated by combined Chaikin algorithm

4.3. Denavit-Hartenberg-algorithm: example 5 & 6

We have also tested the Denavit-Hartenberg-Algorithm. An example is shown in Fig. 14. It illustrates the case where data comes from a ruled quadric. Four lines are chosen and four steps of the refinement based on the Denavit-Hartenberg-Method are applied.

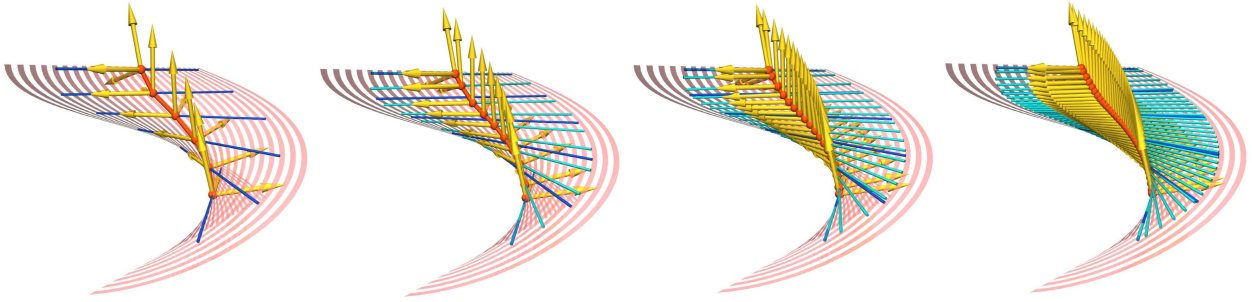


Figure 17: Example 6: Reproducing a helical ruled surface

The third algorithm which uses geodesic subdivision in the group of Euclidean motions also nicely reproduces the shape of surfaces which are invariant under one-parameter subgroups, i.e., helical ruled surfaces. Starting with an initial set of six lines we have performed three steps of the algorithm. The result can be seen in Fig. 17.

4.4. Comparison of algorithms

The projection algorithm does not differ between different types of rulings. Torsal rulings and cylindrical correspond to points in the model space and they can be treated with any averaging procedure. There is no computation of the striction curve and the problems of choosing the right striction point there cannot occur. The adjacent projection does not fail.

Before applying the second algorithm, we have to clarify, whether we want to interpolate or approximate the ruled surface. As outlined in Section 2.4 there are two possibilities for the estimation of the striction point in case of intersecting consecutive rulings. However, any approximating scheme makes sharp corners round. This is also the case for singular rulings of ruled surfaces. Interpolating schemes can be forced to make sharp ridges by reducing weights only locally. This needs a preprocessing step where the possible sharp edges have to be detected and a decision, if they should be reproduced has to be made.

Here we should remark that the discrete version of the striction curve converges quadratically towards the smooth striction curve, if the data lines that come from a smooth surface become finer. Nevertheless the limit curve obtained from a subdivision scheme applied to the previously estimated striction polygon will in general not coincide with the striction curve of a ruled surface that rendered the data lines. It is still a good approximation since it cannot be too far away.

We like to describe a refinement method very similar to the combined algorithm here: Apply a subdivision scheme to the discrete striction curve once and apply the same scheme to the discrete spherical image once. Now we can recompute the discrete striction curve and restart the refinement. In this case the convergence results by WALLNER and DYN [28] cannot be applied. Tests confirm that after a few steps the striction curve as well as the ruled surface shows a somehow chaotic behavior.

Finally we remark that the third algorithm may have the same problems like the combined algorithm since in the first step (computation of the Sannia frames) the discrete version of the striction curve is also computed.

5. Conclusion

We have presented geometric ideas for the approximation and interpolation of ruled surfaces. After some minor modification subdivision schemes for curves nicely apply to ruled surfaces. There is no doubt that these constructions can also be done in the set of spheres in Euclidean space. Geodesic subdivision in the set of spheres is very simple since we can work in the cyclographic model which is in fact an affine space.

For visualization and animation the discrete models of ruled surfaces or canal surfaces are sufficient. It is not necessary to look for parameterizations or representations in terms of implicit equations. In the case of ruled surfaces it is easy to switch to a representation in terms of a triangle mesh (or any other mesh). This will not be the case for canal surfaces when constructed in a similar way. Nevertheless these representations should be sufficiently well for visualizations and animations.

The less elegant approach to ruled surface subdivision would be the following: Consider a discrete ruled surface to be bounded by two polygons. Apply a subdivision to each of the polygons and join the new vertices by lines which then comprise the set of rulings of the refined discrete ruled surface. We did not discuss this method here since it does not present new ideas. It is mainly based on the subdivision of the arbitrarily chosen directing curves. Exactly that is what we tried to circumvent.

Acknowledgements

A lot of thanks go to the reviewers for their helpful comments and hints. They really helped to improve the paper.

References

- [1] O. BOTTEMA, B. ROTH.: *Theoretical kinematics*. Dover Publ., New York 1990.
- [2] G.M. CHAIKIN: *An algorithm for high speed curve generation*. Computer Graphics and Image Processing **3**, 346–349 (1974).
- [3] H.-Y. CHEN, H. POTTMANN: *Approximation by ruled surfaces*. J. Comput. Appl. Math. **102**, 143–156 (1974).
- [4] N. DYN, J.A. GREGORY, D. LEVIN: *A four-point interpolatory subdivision scheme for curve design*. Comp. Aided Geom. Design **4**, 257–268 (1987).
- [5] A. GFRERRER: *On the construction of rational curves on hyperquadrics*. Grazer Mathematische Berichte **340**, 1–69 (2000).
- [6] A. GFRERRER: *Rational interpolation on a hypersphere*. Comp. Aided Geom. Design **16**, 21–37 (1999).
- [7] M.F. HASSAN, I.P. IVRISSIMITZIS, N.A. DODGSON: *An interpolating 4-point ternary stationary subdivision scheme*. Comp. Aided Geom. Design **19/1**, 1–18 (2002).
- [8] J. HOSCHEK: *Liniengeometrie*. Bibliographisches Institut, Zürich 1971.
- [9] M. HUSTY, A. KARGER, H. SACHS, W. STEINHILPER: *Kinematik und Robotik*. Springer, Berlin 1997.
- [10] B. JÜTTLER: *Rationale Bézierdarstellung räumlicher Bewegungsvorgänge und ihre Anwendung zur Beschreibung bewegter Objekte*. PhD Thesis, TU Darmstadt, 1993.

- [11] R. DIETZ, J. HOSCHEK, B. JÜTTLER: *An algebraic approach to curves and surfaces on the sphere and on other quadrics*. Comp. Aided Geom. Design **10**, 211–229 (1993).
- [12] B. JÜTTLER: *Zur Konstruktion rationaler Kurven und Flächen auf Quadriken*. Journal of Geometry **47**, 53–64 (1993).
- [13] S. LEOPOLDSEDER: *Algorithms on cone spline surfaces and spatial osculating arc splines*. Comp. Aided Geom. Design **18/6**, 505–530 (2001).
- [14] M. PETERNELL: *Developable surface fitting to point clouds*. Comp. Aided Geom. Design **21**, 785–803 (2004).
- [15] M. PETERNELL: *Recognition and reconstruction of developable surfaces from point clouds*. Proc. of ‘Geometric Modeling and Processing 2004’, Beijing, China, pp. 301–310.
- [16] PETERNELL, M. POTTMANN, H.: *Interpolating Functions on Lines in 3-Space*. In A. COHEN, Ch. RABUT, L.L. SCHUMAKER (eds.): *Curve and Surface Fitting: Saint Malo 1999*, Vanderbilt Univ. Press, Nashville, TN, 2000, pp. 351–358.
- [17] M. Peterzell, H. Pottmann, B. Ravani: *On the computational geometry of ruled surfaces*. Computer-Aided Design **31**, 17–32 (1999).
- [18] H. Pottmann, M. Peterzell, B. Ravani: *An introduction to line geometry with applications*. Computer-Aided Design **31**, 3–16 (1999).
- [19] H. Pottmann, M. Peterzell, B. Ravani: *Approximation in line space: Applications in robot kinematics and surface reconstruction*. In J. LENARČIČ, M. HUSTY (eds.), *Advances in Robot Kinematics: Analysis and Control*, Kluwer, 1998, pp. 403–412.
- [20] B. ODEHNAL: *Optimization methods for line congruences*. PhD thesis, Vienna Univ. Technology, 2002.
- [21] B. ODEHNAL, H. POTTMANN: *Computing with discrete models of ruled surfaces and line congruences*. Electron. J. Comput. Kinematics 1/1 (2002), §20, Proceedings of the workshop ‘Computational Kinematics’ in Seoul, May 19–22, 2001.
- [22] H. POTTMANN, J. WALLNER: *Approximation algorithms for developable surfaces*. Comput. Aided Geom. Design **16**, 539–556 (1999).
- [23] H. POTTMANN, J. WALLNER: *Computational Line Geometry*. Springer-Verlag, 2001.
- [24] T. DE ROSE, J. STOLLNITZ, D. SALESIN: *Wavelets for computer graphics*. Morgan Kaufmann, San Francisco 1996.
- [25] R. SAUER: *Differenzgeometrie*. Springer-Verlag, 1970.
- [26] K. SHOEMAKE: *Animating Rotation with Quaternion Curves*. Proc. 12th Int. Conf. on Computer Graphics and Interactive Techniques, San Francisco 1985, pp. 245–254.
- [27] H. STACHEL: *Instantaneous spatial kinematics and the invariants of axodes*. Proc. Ball 2000 Symposium, Cambridge 2000, no. 23, 14p.
- [28] J. WALLNER, N. DYN: *Convergence and C^1 -analysis of subdivision schemes on manifolds by proximity*. Comp. Aided Geom. Design **22/7**, 593–622 (2005).
- [29] J. WALLNER, E. NAVA YAZDANI, P. GROHS: *Smoothness properties of Lie group subdivision schemes*. Multiscale Modeling and Simulation **6**, 493–505 (2007).
- [30] J. WALLNER, H. POTTMANN: *Intrinsic subdivision with smooth limits for graphics and animation*. ACM Trans. Graphics **25/2**, 356–374 (2006).
- [31] E.A. WEISS: *Einführung in die Liniengeometrie und Kinematik*. B.G. Teubner, 1935.