Optimal Ruled Surface Fitting

Gershon Elber*

April 23, 2011

Abstract

Ruled surfaces play a major role in many manufacturing processes, such as wire EDM and side CNC milling. In this work, we explore an optimal ruled surface fitting scheme to a given general freeform rational surface, S.

The solution is divided into two parts. In the first and given a line segment that spans two (boundary) points on S, we derive a scheme that efficiently computes the maximal distance from the line segment to S. Then, and using a discrete sampled set of boundary points on S, all the possible line segments between point-pairs are considered and the best ruling fit is derived using dynamic-programming.

The result is an optimal ruling fit for general hyperbolic regions that is less effective on convex domains, a concern we also address in the examples section.

keywords: Hausdorff distance, Dynamic-programming, Surface fitting, Wire EDM, Side CNC milling.

1 Introduction and Previous Work

The manufacturing world shows itself to be quite imaginative when one considers the variety of manufacturing technologies out there. While both 3-axis and multi-axis CNC processes are well known, other manufacturing technologies, such as Wire Electrically Discharged Machining (EDM) [3, 21] and CNC using the *side* of the tool [10, 20, 25] are also in extensive use. In wire EDM, a conducting wire is discharging electricity against a conductive stock and evaporates and cuts ruling lines in the material. Tool side machining exploits the linear edge of the rotating tool to similarly cut along a line[1].

Interestingly enough, other manufacturing processes also perform cuts along lines in 3D. Examples include CNC water jet cutting [2] as well as laser cutters. Recently, the use of hot wire cutting in styrofoam [1] was also proposed towards mold making of surface tiles in architecture designs [19].

With all these manufacturing technologies, it is interesting that no good answer has been given to the following *Ruled Surface Fitting* (RSF) problem:

Problem 1.1 *Given a parametric surface $S(u, v)$, find the* best *ruled surface fit to S, under some norm.*

A solution to Problem 1.1 would clearly increase the usability and quality of all the above mentioned manufacturing technologies, given a surface, S, to be manufactured. The accuracy of the resulting artifact would greatly benefit from such an optimal ruling fit. However, and because a ruled surface can never be elliptical [8], one can expect that only a saddle-like or a hyperbolic patch (or parabolic) S will benefit from such a ruled surface fitting. Forced to use a line-cutting based technology that constructs ruled surfaces, the benefits for elliptical surface regions cannot be significant. Nevertheless, one can still hope to find the optimal ruled surface fit to any surface S, under the constraints of line-cutting manufacturing technologies. In [10], the general surface, S, is recursively divided into strips along isoparametric directions, each of which is fitted with a ruled surface. In [13], a surface division with similar ruled surface fitting goals is made but along isophotes, bounding the normal deviations but not the maximal distance to the ruled fitted surface. In both cases, the computed ruling fit is not optimal. In [15], an approximation of a general surface using ruled surfaces is considered, among other things, and is reduced to solving a non linear optimization function that minimizes distances between the tangent planes of the original surface and the fitted ruled surface, over the domain. Also in [15], the special case of a surface of revolution is considered, exploiting fitted hyperboloids of one sheet that are also ruled surfaces. Some work on fitting ruled surfaces was also done using a curve representation on the Dual Unit Sphere [24, 26].

While not much can be found on optimal ruled surface fitting to general surfaces, there is also a large body of research on piecewise developable surface fitting to general surfaces [5, 9, 17, 18, 22, 23]. Being a sub-class of the ruled surfaces, developable surfaces are also ruled sur-

*Department of Computer Science, The Technion—Israel Institute of Technology, Haifa 32000, Israel. E-mail: gershon@cs.technion.ac.il

[1] While one has to carefully take into account the non-zero thickness of the tool.

faces. In [9, 22], the input general surface is first divided into strips (isoparametric in [9]), which are then fitted with developable surfaces. Yet again, neither case considers optimal fit.

Finally, and while polygonal meshes are not in the scope of this work, the construction of developable sheets for mesh representations has captured much attention as well [17, 18, 23]. In [18], the mesh is divided into polygonal strips and each strip is then laid out flat as a developable sheet. In [17], the mesh model is first decomposed into cylindrical shapes that are then fitting with developable sheets. The work of [23] deals with the best developable fit to a polygonal strip and employs dynamic-programming to establish the optimal match, an approach we similarly employ herein.

This work explores an optimal solution to Problem 1.1. The solution is based on two steps. In the first stage and given a line segment that spans two (boundary) points on S, we present, in Section 2.1, an efficient scheme to compute the maximal distance from the line segment to S. Then, in Section 2.2, we show how the optimal ruling fit can be computed through a discrete set of such sampled line segments. The rest of this work includes examples and possible extensions we present in Section 3, only to conclude in Section 4.

2 Algorithm

In order to offer a viable solution to the RSF Problem 1.1, we discretize the problem. Consider a tensor product parametric surface $S(u, v)$, and without loss of generality let $C_0(u)$ and $C_1(u)$, $u \in [0, 1]$, be the v_{min} and v_{max} boundary curves of S.

Let P_0^i and P_1^j, $i, j \in [0, n-1]$ be $2n$ sampled points along C_0 and C_1, respectively, and denote line L_{ij} as the line segment from P_0^i to P_1^j. In Section 2.1, we portray an algorithm to compute the maximal distance from L_{ij} to surface S,

$$D_H(L_{ij}, S) = \max_{Q \in L_{ij}} \min_{u,v} ||Q - S(u, v)||,$$

which is also the one-sided Hausdorff distance from L_{ij} to S.

With the ability to compute the maximal distance from L_{ij} to S, in Section 2.2, we employ dynamic-programming, over all possible L_{ij}, and find an optimal ruling fit to S, in a quadratic time or $O(n^2)$.

2.1 Line-Surface Distance

Without loss of generality, let $P_0^i = C_0(u_i) = S(u_i, v_{min})$ and $P_1^j = C_1(u_j) = S(u_j, v_{max})$. Denote by l_{ij} the line *in the parametric domain* of S as

$$l_{ij}(t) = (u_i, v_{min})t + (u_j, v_{max})(1 - t), \quad t \in [0, 1],$$

and consider the composition [7] of

$$\mathcal{L}_{ij}(t) = S(l_{ij}(t)), \quad t \in [0, 1].$$

See also Figure 1.

In $\mathbb{R}^3$, L_{ij} is clearly a straight line. However, $\mathcal{L}_{ij}(t)$ is not and the deviation of $\mathcal{L}_{ij}(t)$ from L_{ij} provides an upper bound on the maximal distance from L_{ij} to S.

Without loss of generality, we further simplify the problem by applying rigid motion, $\mathcal{R}$, to this arrangement, bringing it into a canonical position so that P_0^i is at the origin and L_{ij} unites with the $+Z$ axis. Then, $\mathcal{L}_{ij}(0) = \mathcal{R}(P_0^i) = \vec{0}$ and $\mathcal{L}_{ij}(1) = \mathcal{R}(P_1^i)$ on the $+Z$ axis. Hereafter and for the rest of this section, we assume that the surface is in a canonical position.

In the canonical position, the maximum deviation of $\mathcal{L}_{ij}(t)$ from L_{ij} is the maximal XY radial deviation of the XY projection of $\mathcal{L}_{ij}(t)$ from the origin, and hence is the solution of

$$\left\langle \mathcal{L}_{ij}^{xy\prime}(t), \mathcal{L}_{ij}^{xy}(t) \right\rangle = 0, \tag{1}$$

where $\mathcal{L}_{ij}^{xy}(t)$ is the projection of $\mathcal{L}_{ij}(t)$ on the XY plane. Figure 1 shows one example of an XY radial deviation of $\mathcal{L}_{ij}(t)$ from the origin, along with the extremal location, which serves as an upper bound on D_H, as $\lceil D_H(L_{ij}, S) \rceil$.

Having established a tight upper bound on $D_H(L_{ij}, S)$, $\lceil D_H(L_{ij}, S) \rceil$, we are now ready to derive its precise value. Recall that line L_{ij} is above the origin, in the $+Z$ direction. The extrema of any general function, and the distance function specifically, can be:

1. At the end points–but here the end points of L_{ij} are selected to be on S.
2. At local extrema locations or at the *antipodal locations* where the antipodal line, denoted $\mathcal{A}(D_H)$, is orthogonal to both L_{ij} and S, at the two points where $\mathcal{A}(D_H)$ intersects L_{ij} and S.
3. At discontinuities in the function. Here C^1 discontinuities can occur in the distance function along the self bisector of S [4, 11]. However, the fitted ruled surface and S are expected to be much closer compared to the magnitudes of the radii of curvatures of S. In other words, the bisector sheet of S is expected to be much further away from S compared to the distances to the fitted ruled surface. Hence, this third case is ignored for the rest of this work.

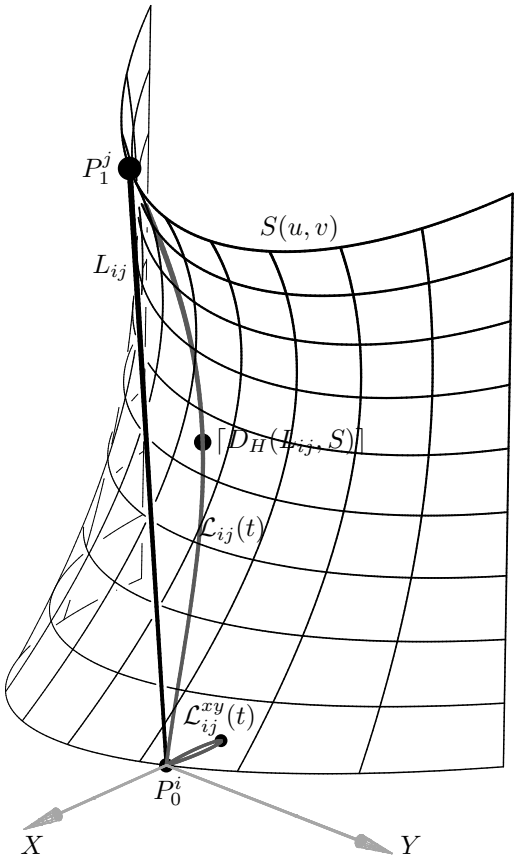


Figure 1: An upper bound on the maximal deviation of line L_{ij} (in black) from surface S is established via the distance to the composition $\mathcal{L}_{ij}(t) = S(l_{ij}(t))$ (in gray), where l_{ij} is the line connecting P_0^i and P_1^j in the parametric domain of S. Being in a canonical position, L_{ij} is along the Z axis and hence this upper bound can be established on the XY projection of $\mathcal{L}_{ij}(t)$, $\mathcal{L}_{ij}^{xy}(t)$.

We now focus on the second extrema case of *antipodal locations*. At an *antipodal location* extrema of the distance between $L_{ij}(t)$ and $S(u,v) = (X(u,v), Y(u,v), Z(u,v))$ in the canonical position, the following equations must hold:

$$\begin{aligned} N_z(u,v) &= 0, \\ N_x(u,v)Y(u,v) - N_y(u,v)X(u,v) &= 0, \end{aligned} \quad (2)$$

where $N(u,v) = (N_x(u,v), N_y(u,v), N_z(u,v))$ is the normal field of $S(u,v)$,

$$N(u,v) = \frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v},$$

and N is well defined for a regular surface S.

Recall $\mathcal{A}(D_H)$ is the line between the two *antipodal locations* on $L_{ij}(t)$ and $S(u,v)$ respectively, where D_H holds. Because line $L_{ij}(t)$ is vertical or along $+Z$, and since $\mathcal{A}(D_H)$ must be orthogonal to both shapes, $\mathcal{A}(D_H)$ must be horizontal, or $N_z(u,v) = 0$, which is the first constraint in Equations (2), Then, the orthogonality to S is ensured by the second constraint of Equations (2).

Having two equations and two unknowns, Equations (2) are fully constrained. Nonetheless, a general surface can have numerous locations where Equations (2) are satisfied. In order to streamline the process and make it more efficient, we only consider solutions within the established upper bound, $\lceil D_H(L_{ij},S)\rceil$, such that $||L_{ij}(t) - S(u,v)|| \leq \lceil D_H(L_{ij},S)\rceil$. Out of this validated solution set, the largest computed distance that is less than or equal to $\lceil D_H\rceil$ is set as the precise D_H between L_{ij} and S.

Equation (1) and Equations (2) are solved using the multivariate piecewise-polynomial solver of [12, 14]. In fact, the inequality constraint of $||L_{ij}(t) - S(u,v)|| < \lceil D_H(L_{ij},S)\rceil$ is embedded in the solution process, eliminating any need to process regions in $S(u,v)$ that are far from $L_{ij}(t)$.

2.2 Best ruling fit

Having the ability to precisely derive $D_H(L_{ij},S)$, $\forall i,j$, we are now ready to derive the optimal ruling possible, for S. Assume S is open and hence the first ruling line must be between $P_0^0 = (u_0, v_{min}) \in C_0$ and $P_1^0 = (u_0, v_{max}) \in C_1$ and is L_{00}. Similarly, and recalling we sampled n samples in each boundary, the last ruling line must be $L_{n-1,n-1}$. Consider now the intermediate ruling line between P_0^i and P_1^j, L_{ij}, $i,j > 0$. The previous ruling line to L_{ij} can be one of

$$L_{i-1,j}, \;\; L_{i,j-1}, \;\; \text{or} \;\; L_{i-1,j-1}, \quad (3)$$

meaning, a ruling line that is previous in i, or previous in j, or both.

With Equation (3) and the initial condition of L_{00}, L_{i0}, $i > 0$, can have only one previous ruling line which is $L_{i-1,0}$. Similarly, the only possible previous ruling line to L_{0j}, $j > 0$, is $L_{0,j-1}$.

This recursive computational view is also known as *dynamic-programming* [6]. Denote the distance of ruling line L_{ij} to S by D_H^{ij} and let $\sum_{ij} D_H$ be the minimal sum of all distances of ruling lines to S, in some path, from L_{00} to L_{ij}. A 2D table can be used to represent the dynamic-programming process:

$$\begin{array}{ccccc} \sum_{00} D_H & \cdots & \sum_{i0} D_H & \cdots & \sum_{n-1,0} D_H \\ \vdots & & \vdots & & \vdots \\ \sum_{0j} D_H & \cdots & \sum_{ij} D_H & \cdots & \sum_{n-1,j} D_H \\ \vdots & & \vdots & & \vdots \\ \sum_{0,n-1} D_H & \cdots & \sum_{i,n-1} D_H & \cdots & \sum_{n-1,n-1} D_H, \end{array}$$

where, and following Equation (3), we have,

$$\begin{aligned}
\sum_{00} D_H &= D_H^{00}, \\
\sum_{i0} D_H &= D_H^{i0} + \sum_{i-1,0} D_H, \\
\sum_{0j} D_H &= D_H^{0j} + \sum_{0,j-1} D_H, \\
\sum_{ij} D_H &= D_H^{ij} + \\
&\min\left(\sum_{i-1,j} D_H, \sum_{i,j-1} D_H, \sum_{i-1,j-1} D_H\right).
\end{aligned} \tag{4}$$

Clearly, all the entries of the 2D table can be computed in $O(n^2)$, filling the table row by row, top to bottom and left to right. Furthermore, over the input sampled set of points $P_{kj},\ k = 0, 1,\ j = [0, n-1]$, the computed dynamic-programming solution is globally optimal. Rulings that globally minimize the sum of all deviations between ruling lines and S can hence be computed in $O(n^2)$. Note that we do not minimize the maximal error but rather minimize the total (sum) error.

One can prescribe an additional constraint for the maximal allowed distance between any ruling line and S, as D_H^{valid}, coercing every $D_H^{ij} > D_H^{valid}$ to be $D_H^{ij} \to \infty$, rendering D_H^{ij} viable in no solution. It should be recalled that this additional D_H^{valid} constraint can result in no viable solution at all, having no path with a finite cost from L_{00} to $L_{n-1,n-1}$ in the dynamic-programming table.

To recover the matching of the ruling with the minimal cost, one should back-track, starting from location $(n-1, n-1)$ in the table and see which of $(n-2, n-1)$, $(n-1, n-2)$ and $(n-2, n-2)$ was the predecessor of $(n-1, n-1)$, continuing this back-tracking process until reaching position $(0, 0)$.

This back-tracking process creates a list of pairs, $(0, 0)$, ..., (i, j), ..., $(n-1, n-1)$, which is monotone (but not strictly monotone) in both axes. A scalar function, $f(t)$, is then fitted to this sequence such that $f(0) = 0$, $f(i) = j$ and $f(n-1) = n-1$. While the problem of spline fitting to sample points is beyond the scope of this work and many approaches can be used here, it should be recalled that because the pairs' list is not strictly monotone, f can be non-regular as a result. Some minimum positive speed must be assigned to f to ensure it is also regular. Thereafter, an optimal ruled surface is defined between $C_0(u)$ and $C_1(u)$ as

$$R(u, v) = C_0(\hat{f}(u))v + C_1(u)(1-v),\ \ u, v \in [0, 1], \tag{5}$$

where $\hat{f}(u)$ in Equation (5) equals,

$$\hat{f}(u) = \frac{f((n-1)u)}{n-1},$$

mapping the $[0, n-1]$ domain and range of f back to $[0, 1]$. Also one should note that the construction of R in Equation (5) necessitates the computation of the curve-curve composition, $C_0(\hat{f}(u))$ [7].

The next section presents some examples and also discusses several immediate extensions to the presented approach.

3 Examples and Extensions

All examples presented in this work were created using an implementation that is part of the IRIT [16] modeling environment, taking only a few seconds to compute on a modern PC workstation. Figures 2 and 3 show two simple open hyperbolic surfaces, Bézier and B-spline respectively, which are quite well fitted with a ruling approximation.

In open surfaces, the simplest ruled surface matching process examines points in one boundary against points in the opposite boundary. However, if the surface is closed or periodic, the matching can wrap around the closed boundary, offering a better match in many cases.

In order to support closed and/or periodic surfaces in the dynamic-programming algorithm, all we need do is allow a shift in the initial search from ruling line L_{00} to L_{i0}, $\forall i \in [0, n-1]$ and terminate it in ruling line $L_{i-1,n-1}$. Then, we also allow periodic behavior in the table. That is, we allow $D_H^{0,j}$ to also depend on $D_H^{n-1,j}$, etc. If building one such table costs $O(n^2)$, building n such tables, each starting at ruling line L_{i0}, $i = 0, ..., n-1$, will cost $O(n^3)$ in all. Once all n tables are built, their best match is selected as the optimum.

In Figure 4, a periodic hyperbolic surface that is a surface of revolution of a conic section is fitted as a ruled surface. Our algorithm that was extended to support periodic surfaces, was capable of recovering this result quite accurately as a ruled surface and offering a parametrization that reconstructs the original surface to a great extent.

Figure 5 shows the two parametrization functions, $\hat{f}(u)$, reconstructed for the examples in Figures 3 and 4. The example in Figure 5 (b), for the surfaces in Figure 4, is periodic; hence, it wraps around the v domain of the input surfaces.

Figure 6 show a hyperbolic surface that is not a surface of revolution. In this case the use of hyperboloids as suggested in [15] would yield poor results. Our approach, however, is still able to recover this shape with great accuracy, as a ruled surface. Finally, Figure 7 presents a

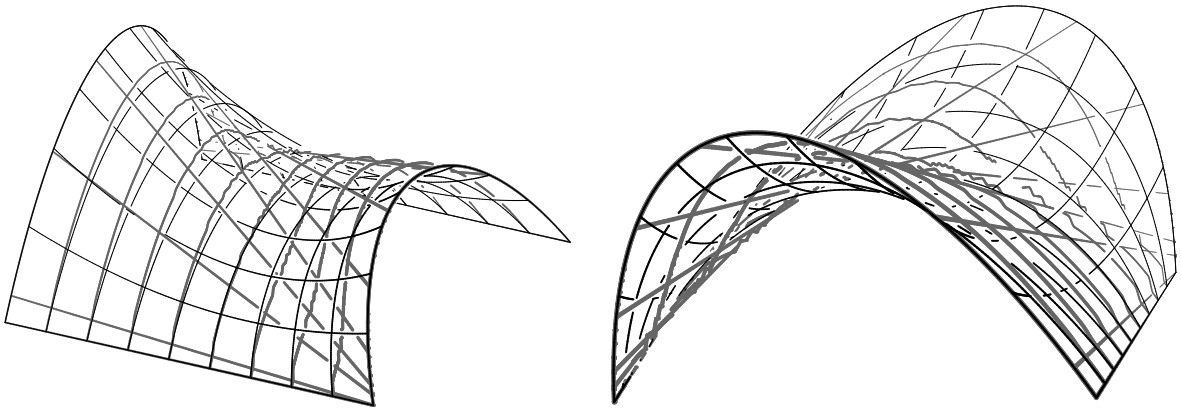


Figure 2: Two views of a quadratic by cubic hyperbolic Bézier patch (in black) along with the optimal fit ruled surface (in gray), computed using the presented algorithm.

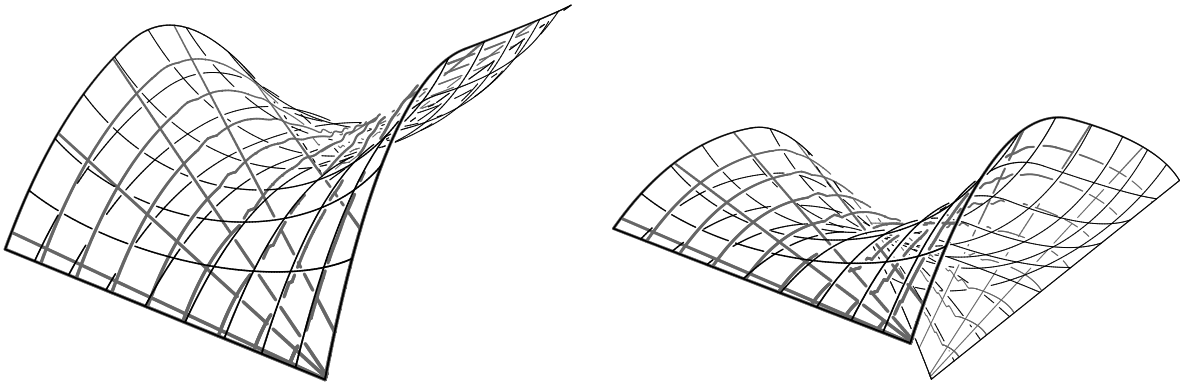


Figure 3: Two views of a quadratic by cubic hyperbolic B-spline surface of mesh of size (4×4) (in black) along with the optimal fit ruled surface (in gray), computed using the presented algorithm. (see also Figure 5).

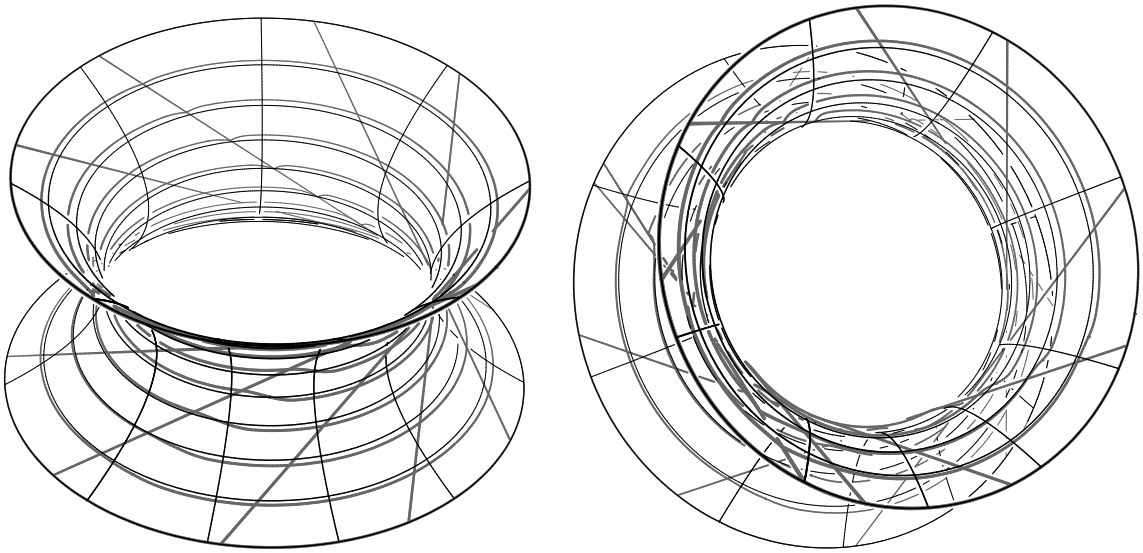
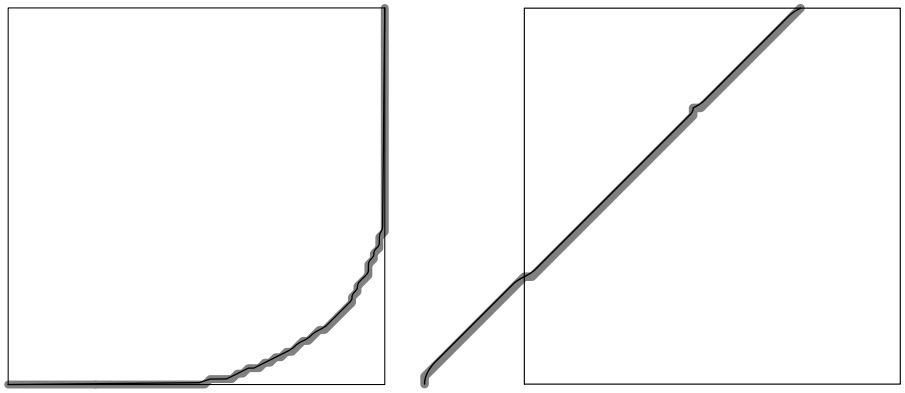


Figure 4: Two views of a quadratic by cubic hyperbolic B-spline surface of mesh of size (3×13) (in black) along with the optimal fit ruled surface (in gray), computed using the presented algorithm. This surface is periodic and the optimal ruling parametrization indeed crosses the boundary (see also Figure 5).



(a) (b)

Figure 5: The reparametrization functions, $\hat{f}(u)$ (Equation (5)), of the examples in Figure 3, in (a), and Figure 4, in (b). $n = 70$ sample points were used in (a) for each curve and $n = 50$ in (b). The discrete result of the dynamic-programming is shown in thick gray, whereas the thin dark curve is the quadratic fit, $\hat{f}(u)$. The square presents the original surface domain, $(u \times u)$. Note that (b) is a periodic case, in which the best match is found after a shift is applied to the initial dynamic-programming setup, resulting in a parametrization that wraps around the periodic boundary.

long strip of a surface that is convex. Nonetheless, our algorithm is still able to offer a somewhat better ruling parametrization compared to the input parametrization, as can be seen by the diagonal ruling result, in gray.

The reduced error, both in terms of the total sum of all ruling directions and in the maximum distance, D_H, senses, is very significant. In all hyperbolic cases and recalling that these hyperbolic shapes are not precise ruled surfaces, the errors range from a half and down to about 10% of the original total and maximum errors, compared to a naive direct ruling of $C_0(u)v + C_1(u)(1-v),\ u, v \in [0,1]$. Even in the case of the elliptic (convex) surface (Figure 7), some improvement in the total error of about 5% was gained due to the optimally selected ruling reparametrization.

4 Conclusions and Future Work

In this work we presented a method to derive an optimal ruled surface fitting, among all sampled ruling locations. Different sampling approaches of points, other than on two opposite boundary curves of S, can yield different and even better results. Clearly, given a general surface, S, one can aim at a ruling fit to the u boundaries and then to the v boundaries, selecting the better of the two. More interesting is the possible use of *interior* isoparametric lines as the sources for the sampled points. Such a selection can yield superior results as the error is going to be spread into the two opposite directions, both below and above the ruled surface fit. For example, see Figure 8 where both the maximal and total errors are roughly halved due to using this approach.

Consider a half of the surface of revolution in Figure 4, of 180 degrees out of the full surface of revolutions; see also Figure 9. Any attempt to fit a ruled surface through the boundaries of this open surface is futile, as can be seen in Figures 9 (b) and (c). The proper solution will allow the *extension of the two sampled boundaries* beyond their end points only to trim the ruled surface to the domain of the input surface S. This extension can be implemented in two different ways:

1. Given a C^k boundary curve $C(u),\ u \in [0,1]$ build a new C^k extended curve $C_e(u),\ u \in [-a, 1+a], a > 0$ such that $C(u) = C_e(u),\ u \in [0,1]$. Once the ruled surface is built for $C_e(u)$, it will be trimmed back to the domain of the original surface S.

2. Without loss of generality and given boundary curve $C_0(u) = S(u, v_{min})$, match the points on $C_0(u)$ against points on the three other boundary curves of S in order, starting with points on $S(u_{min}, v)$, then on $S(u, v_{max})$, and then on $S(u_{max}, v)$. This matching will be performed for every boundary curve (against the three other boundary curves), four times in all, and then the best one is selected.

Method 2 can be seen as a special case of method 1 where $C(u) = S(u, v_{max})$, is extended before by curve $S(u_{min}, v)$ and after by reversed curve $S(u_{max}, v)$.

Beyond this augmented sampling scheme that can resolve the problem portrayed in Figure 9, different, more preferred sampling of points on the surfaces, toward ruled surface fitting, should be examined.

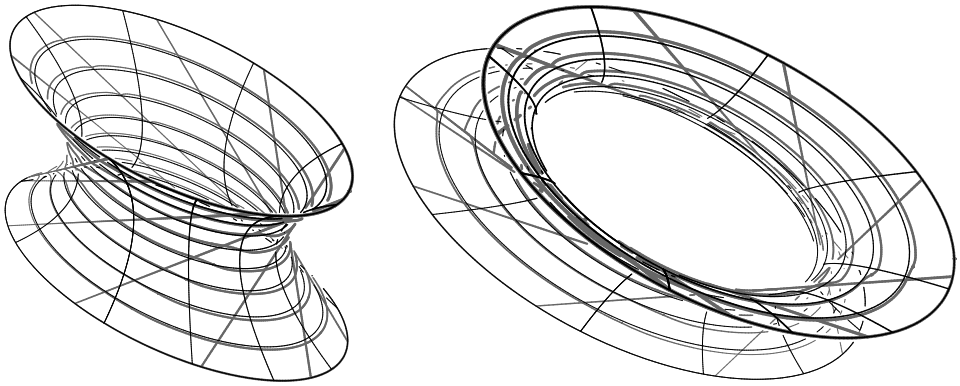


Figure 6: Two views of an example of a non-surface of revolution periodic case where hyperboloids' fitting cannot be effectively employed. Yet, the presented algorithm recovers the shape, as a ruled surface, with high accuracy.

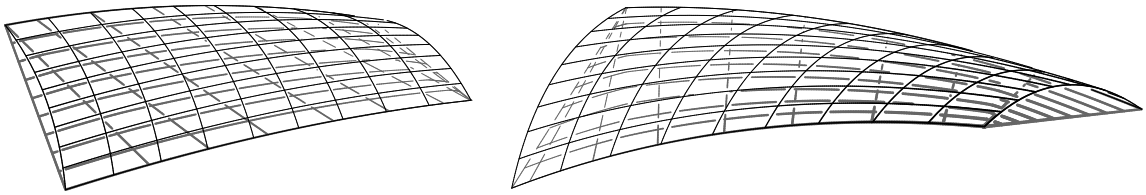


Figure 7: Two views of an example of a elliptic (convex) strip surface. The optimal ruling is not using the input surface parametrization but rather a bit diagonal one, which imrpoves the ruling fit by about 5%.

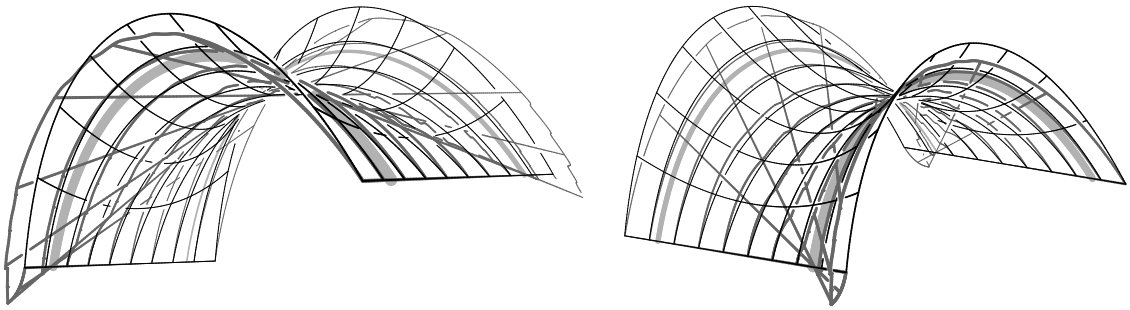
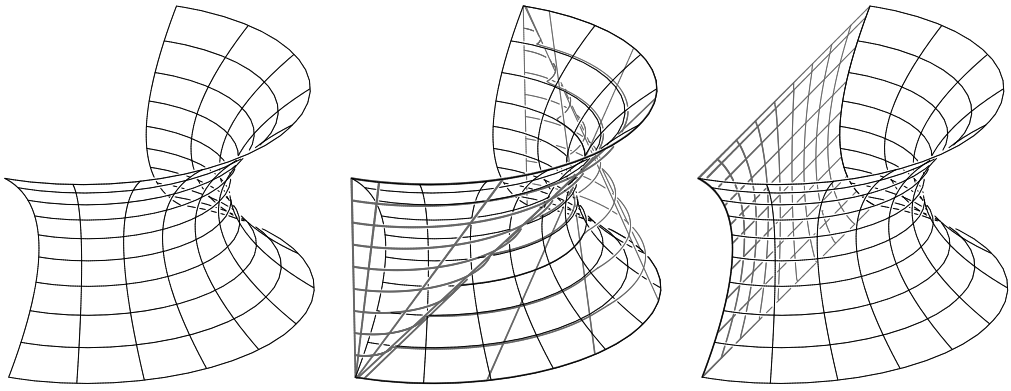


Figure 8: By selecting two *interior* isoparametric curves (thick light gray) and fitting a ruled surface through them, the total error can be reduced, spreading it to the opposite sides of the created ruled surface.



(a) (b) (c)

Figure 9: The input surface in (a) is part of a ruled surface and yet by examining only the u boundaries (b) and/or only the v boundaries (c), the end ruling fit will be quite poor.

References

[1] Hot wire styrofoam cutting. http://en.wikipedia.org/wiki/Hot-wire_foam_cutter.

[2] Water jet cutter. http://en.wikipedia.org/wiki/Water-_jet_cutter.

[3] Wire electrically discharged machining. http://en.wikipedia.org/wiki/Electrical_discharge_machining.

[4] ALT, H., AND SCHARF, L. Computing the hausdorff distance between curved objects. *International Journal of Computational Geometry and Applications 18* (June 2008), 307–320.

[5] CHENA, H. Y., LEE, I. K., LEOPOLDSEDER, S., POTTMANN, H., RANDRUP, T., AND WALLNER, J. On surface approximation using developable surfaces. *Graphical Models and Image Processing 61* (1999), 110–124.

[6] CORMEN, T. H., LEISERSON, C. E., AND RIVEST, R. L. *Introduction to Algorithms*. The MIT Press, 2009.

[7] DEROSE, T. D., GOLDMAN, R. N., HAGEN, H., AND MANN, S. Functional composition algorithms via blossoming. *ACM Trans. Graph. 12* (April 1993), 113–135.

[8] DOCARMO, M. P. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[9] ELBER, G. Model fabrication using surface layout projection. *Computer Aided Design 27* (1995), 283–291.

[10] ELBER, G., AND FISH, R. 5-axis freeform surface milling using piecewise ruled surface approximation. *Journal of Materials Processing Technology 119* (1997), 383–387.

[11] ELBER, G., AND GRANDINE, T. Hausdorff and minimal distances between parametric freeforms in r^2 and r^3. In *Proceedings of Advances in Geometric Modeling and Processing* (2008), GMP 2008, Lecture Notes in Computer Science 4975, Springer, pp. 191–204.

[12] ELBER, G., AND KIM, M. S. Geometric constraint solver using multivariate rational spline functions. In *Proceedings of the sixth ACM symposium on Solid modeling and applications* (New York, NY, USA, 2001), SMA '01, ACM, pp. 1–10.

[13] HAN, Z., YANG, D. C. H., AND CHUANG, J. J. Isophote-based ruled surface approximation of freeform surfaces and its application in nc machining. *International Journal of Production Research 39* (2001), 1911–1930.

[14] HANNIEL, I., AND ELBER, G. Subdivision termination criteria in subdivision multivariate solvers using dual hyperplanes representations. *Comput. Aided Des. 39* (May 2007), 369–378.

[15] HOSCHEK, J., AND SCHWANECKE, U. Interpolation and approximation with ruled surfaces. In *The Mathematics of Surfaces VIII* (1998), Information Geometers, pp. 213–231.

[16] IRIT. The irit geometric modeling environment 2010. http://www.cs.technion.ac.il/∼irit.

[17] MASSARWI, F., GOTSMAN, C., AND ELBER, G. Paper-craft from 3d polygonal models using generalized cylinders. *Computer Aided Geometric Design 25* (2008), 576–591.

[18] MITANI, J., AND SUZUKI, H. Making papercraft toys from meshes using strip-based approximate unfolding. In *ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), pp. 259–263.

[19] POTTMANN, H. Private communication.

[20] REDONNET, J. M., RUBIO, W., AND DESSEIN, G. Side milling of ruled surfaces: Optimum positioning of the milling cutter and calculation of interference. *International Journal of Advanced manufacturing Technology 14* (1998), 459–465.

[21] SPEDDING, T. A., AND WANG, Z. Study on modeling of wire edm process. *Journal of Materials Processing Technology 69* (1997), 18–28.

[22] SUBAG, J., AND ELBER, G. Piecewise developable surface approximation of general nurbs surfaces, with global error bounds. In *GMP'06* (2006), pp. 143–156.

[23] TANG, K., AND WANG, C. C. L. Modeling developable folds on a strip. *Journal of Computing and Information Science in Engineering (ASME Transactions 5* (2005), 35–47.

[24] ZHA, X. F. A new approach to generation of ruled surfaces and its applications in engineering. *International Journal of Advanced manufacturing Technology 13* (1997), 155–163.

[25] ZHANG, X. M., ZHU, L. M., ZHENG, G., AND DING, H. Tool path optimisation for flank milling ruled surface based on the distance function. *International Journal of Production Research 48* (2010), 4233–4251.

[26] ZHOU, Y., SCHULZE, J., AND SCHÄFFLER, S. Blade geometry design with kinematic ruled surface approximation. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (New York, NY, USA, 2010), ACM, pp. 1266–1267.