

Research on Developable Surface Modeling and the Processing Technology

Jun Tang

*Department of Communication Engineering, Xiamen University of Technology,
Xiamen, Fujian 361024, China
E-mail: xmtangjun@126.com*

Abstract

A developable surface is a surface that can be flattened onto a plane without distortion, neither in angle nor in area. In mathematics, it is a collection of points with zero Gaussian curvature. The developable surfaces are of vital importance in modern industry modeling, whereas current business modeling software could only transfer a 3D model into an approximation of a developable surface. In this way, a methodology that could directly generate developable surfaces with no intermediate transformation is in need. This paper illustrated a modeling method based on dynamic programming, and further proposed a method for variational developable surface design. Finally, algorithm in this paper would be verified as valid and practicable by some design cases.

Keywords: *developable surface, graphics and image processing, graphic modeling, dynamic programming.*

1. Introduction

A developable surface is a surface that can be flattened onto a plane without stretching or compressing [1]. Recently, research fellows have been concentrating on the design of developable surface for years. Roles of developable surface in production and daily life, such as building design, clothing manufacturing, lathe machining, etc., are absolutely significant. Otherwise, in the field of computer graphics and computer-aided design (CAD) including origami, architecture and industrial surface design, the developable surface is also inevitable.

Discrete developable surfaces are pieces of grid developable surface, with the shape of triangle or quadrangle, who subdivide the measure of developable surface ultimately [2]. As the concern of practical software like CAD arose, studies on discrete developable surfaces focus on the design of triangular or quadrangular developable grids that could satisfy the developable measures exactly or approximately [3]. Developable surfaces modeling could be divided into two categories- approximate modeling and extract modeling. These two modeling categories is to maximize the develop ability of surface, but the former method does not reach a final model that strictly under constrains, while the later one does. At first, this paper demonstrated an extract modeling method for developable surfaces [4]. By observing its recursive nature, an algorithm with dynamic programming was constructed.

2. Developable Surface Modeling Techniques

2.1. Dynamic programming method

In this section, an interpolation method of developable surface on polygon boundary was introduced based on the theory of dynamic programming. As can be proved that a developable surface must be a ruled surface simultaneously, a

common method for developable surface interpolation is boundary triangularization [5-8]. But the boundary triangularization grow its patterns exponentially with the number of sample points on boundary, it could be infeasible in practice. Due to the recursive nature of dynamic programming, this study presented a global optimal solution for calculation [9]. This solution makes the complicated design of developable surface come true by simplifying calculation process.

Suppose that P is a closed polygon who approximating a closed curve in a three-dimensional space. Its vertexes are recorded anti-clockwise by $P_i(i=1,2,\dots,m)$. Line segment P_iP_j is short donated as $\langle i,j \rangle$ and named rung. Without considering the endpoints, any line segment $\langle i,j \rangle$ is absolutely involved in polygon P (shown in figure 1-(a) and (b)) [6]. A rung consisting of points with a same tangent plane is called an untwisting rung. If all rungs of a grid surface are untwisted, this surface is called a discrete developable surface. In any rungs $\langle i,j \rangle$, we generally set $i \leq j$. The twisting degree of any rungs $l = \langle i,j \rangle$ is defined to be $w(l)$:

$$w(l) = 1 - \frac{t_i \times (P_j - P_i)}{\|P_j - P_i\|} \cdot \frac{t_j \times (P_i - P_j)}{\|P_i - P_j\|} \quad (1)$$

In this equation, t_i, t_j represents the unit tangent vector of P_j, P_i respectively. For any triangularization T of P , $W(T) = \sum w(e)$ is the total twist of T , while e is a side of T . Specially, $D(P)$ represents the minimum of all $W(T)$ values, and the corresponding triangularization T^* is the optimal triangularization of P .

Since P is a simple polygon, it could always be divided into two parts by a rung $l = \langle i,j \rangle$. Setting $P(l^+)$ represents the half involving vertex P_{i+1} , while $P(l^-)$ is the other half [7]. According to the common rule of C language, $i+$ means $i+1$, $i-$ means $i-1$. Supposing rung l is one of three sides of the optimal triangularization of P , and it is the optimal rung. Then the equation below is true [8]

$$D(P) = D(l^+) + D(l^-) + w(l) \quad (2)$$

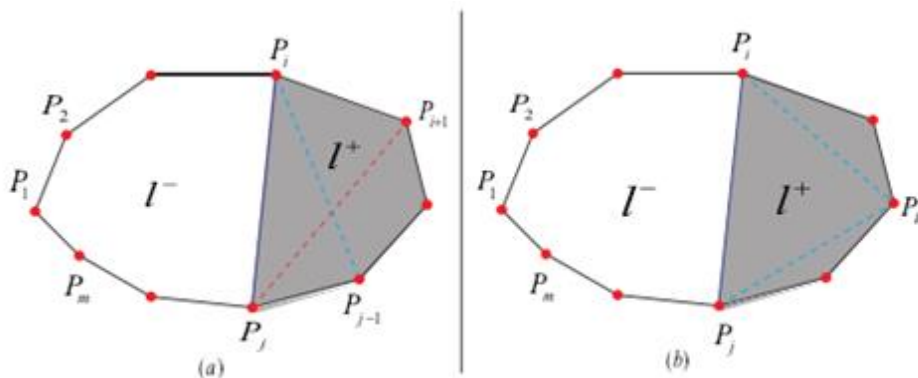


Figure 1. The Polygon is Divided into Two Parts of $P(i+)$ and $P(i-)$ by a rung $\langle i,j \rangle$.

- (a) Two bridge triangles of $\langle i,j \rangle$**
- (b) Visible point P_k of rung $\langle i,j \rangle$**

For each vertex P_i , $L(i)$ indicates a rung set rooting in P_i . Generally speaking, $|L(i)| \leq m-3$. Any triangularization of P inevitably include either a range in $L(i)$ or $\langle i-, i+ \rangle$. By applying equation (2) with each rung in $L(i)$ and $\langle i-, i+ \rangle$, a minimum $D(P)$ could be found, and the corresponding triangularization is the optimal one. To avoid exponential processing time, this study adopted a guideline of dynamic programming presenting in (3) and (4).

$$D\left(P\left(\langle i, j \rangle^+\right)\right) = \text{Min}\left\{D\left(P\left(\langle i+, j \rangle^+\right)\right) + w(\langle i+, j \rangle), D\left(P\left(\langle i, j- \rangle^+\right)\right) + w(\langle i, j \rangle)\right\} \quad (3)$$

$$D\left(P\left(\langle i, j \rangle^-\right)\right) = \text{Min}\left\{D\left(P\left(\langle i+, j \rangle^-\right)\right) + w(\langle i-, j \rangle), D\left(P\left(\langle i, j- \rangle^-\right)\right) + w(\langle i, j \rangle)\right\} \quad (4)$$

Now introduce a similar recurrence algorithm to obtain the optimal triangularization. In this term, three sides of a triangle could be all rungs. To get the goal, an assumption should be made that there exists a plane B that any triangularizations of p are monotonic about it. In addition, any vertexes P_k are supposed to be visible for a rung if $\langle i, k \rangle, \langle k, j \rangle$ are all rungs. Thus, the triangle $P_i P_k P_j$ is totally involved in P except three vertexes. Let $V_{(i)}$ be the collection of all visible rungs for P_i , we get an accordingly tenable rule for recurrence as follows:

$$D\left(P\left(\langle i, j \rangle^+\right)\right) = \text{Min}\left\{D\left(P\left(\langle i, k \rangle^+\right)\right) + w(\langle i, k \rangle), D\left(P\left(\langle k, j \rangle^+\right)\right) + w(\langle k, j \rangle)\right\}; k \in V(i) \cap V(j) \quad (5)$$

Apparently, the set L for any vertexes is still valid. Applying equation (2) on each rung in L_i , combining the recurrence rule (5), an optimal triangularization for p could be obtained.

As for a multiply connected domain bounded by several curves (Figure 2(a)), one or more artificial cut (Figure 2(b)), or several optimal segments connecting vertexes of the surface, could transfer it into some simply connected regions with ease. In this way, methods mentioned above could be valid to find the optimal triangularization for each simply connected region. Thus, the optimal triangularization for a multiply connected domain is completed.

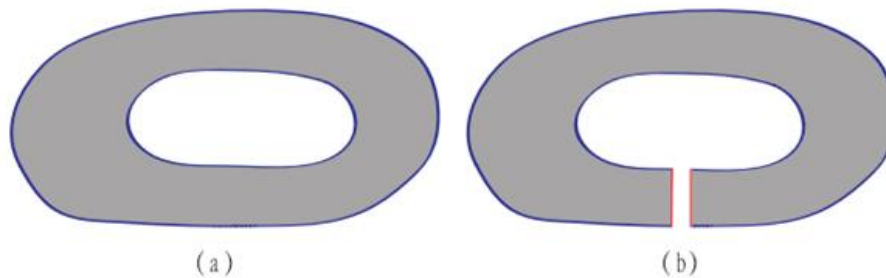


Figure 2. A Multiply Connected Domain in (a) and an Artificial Cut in (b)

2.2. Variational Interpolation Methods

Interpolation and approximation are central questions of the Computer Aided Geometric Design (CAGD). To make it simple, interpolation requires a strict pass of a surface through the given interpolation point, while approximation does not. But the Variational Developable Surface Interpolation (VDSI) is different from those two. For one thing, it claims the surface to approximate some interpolation points within a certain region. For another, it needs the surface to go through the remaining points at the same time. In other word, the VDSI is a hybrid of both interpolation and approximation.

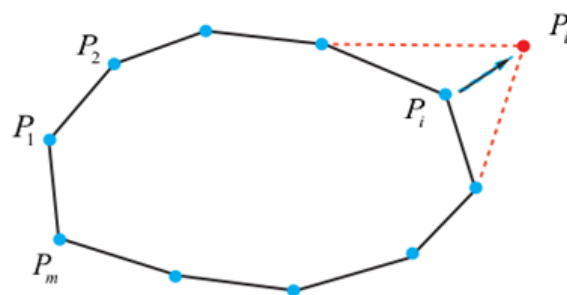


Figure 3. Modify the Polygon by Adjusting Vertex Positions within Limits. Pi' Stands the Adjusted Position of Pi

Suppose that C is a simply smooth closed boundary curve approximated by a closed polygon p . As shown in figure 3, the smooth boundary curve C is approximated by a polygon $P = \{P_1, P_2, \dots, P_m\}$, some of whose vertexes, like P_i are removable within certain region. One important goal of this section is to find a developable surface that is more proper to interpolate a given boundary curve [10]. So the twisting degree $w(l)$ of rung i could be treated as a function of some vertexes.

If the new positions of moveable points P_1, P_2, \dots, P_n ($n \leq m$) are recorded as P_1', P_2', \dots, P_n' ($n \leq m$), within the appointed boundary \in , the optimization function is:

$$\begin{aligned}
 W(T^*) &= \text{Min} \sum_{l \in T} w(l) \\
 \sum_{i=1}^n \|P_i - P_i'\| &\leq \in \\
 P_j - P_j' &(j = n+1, \dots, m.)
 \end{aligned}
 \tag{6}$$

In which T is any triangularization of the points set.

3. Experimental Results and Illustration

This section first introduced the modeling system according to dynamic programming for designing a developable surface. Processing design with flexible interactions is more convenient to developable surface modeling with dynamic programming. Purpose of this modeling system is to present a simple and easy tool that the non-professionals could apply to make developable surface models they need in practice. The modeling system is shown in Figure 4. It can be seen that the user-designed closed curves are presented on the left half of system screen. They are poly-line presentations of the automatically recorded moving paths of a brush or a mouse. Then the presentations would be simplified with Douglas-Pucker methods

and approximated by a sample line B. This tool is so practicable that non-professional users could edit curves, say insert or remove points, easily by handling control point. Thus a multiply connected domain is graphed. On the right half part of screen, there is a render window for space R in which users could move the vertexes for further adjustment. However, adjustment here is restricted to along the vertex normal only. Specially, the 2D curve projection on x-y plane is exactly that drew by users. Adjustment in both half screens would be continually operated until a satisfactory image presents. Once the boundary curves were designed, the dynamic programming methods introduced before would come into effect to interpolate the discrete developable surface.

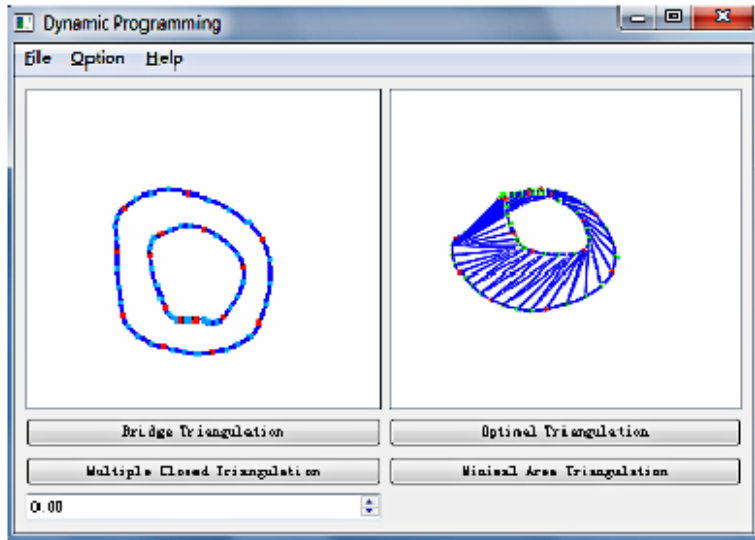


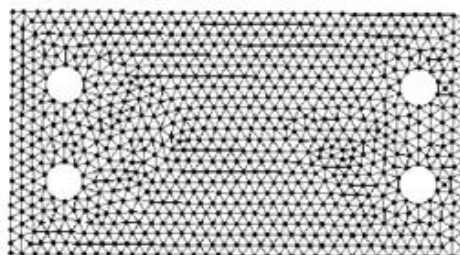
Figure 4. The Developable Surface Modeling System

3.1. Developable Surface Design for Multiply Connected Domains

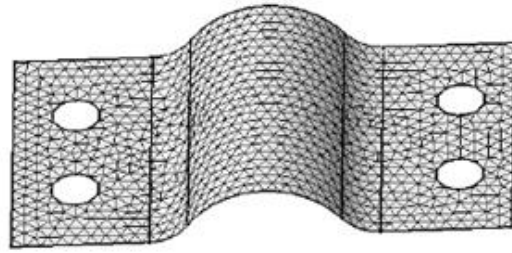
There is a multiply connected region p on a special surface. The optimal ruled surface is drawn by interpolating a developable surface piece on p using methods mentioned above. These methods tend to minimize the change of tangent plane, and the optimal triangularization is not restricted as a bridge one [11]. Corresponding Gauss image demonstrates that surfaces rebuilt by methods in this paper shows better developability than the commonly rebuilt ones.

3.2. Design for Composite Developable Surfaces

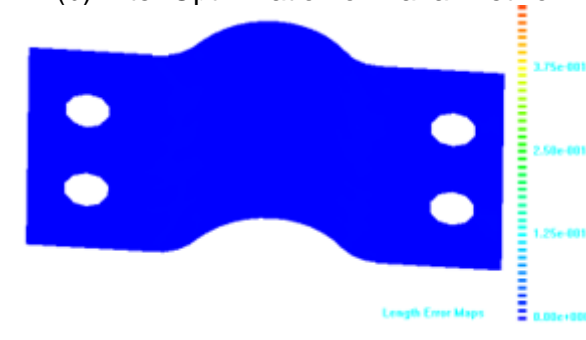
Methods in this paper could reconstruct developable surface by interpolating certain given boundary curves. By this way, composite developable surfaces may be constructed as follows.



(a) Original Surface Grid Model

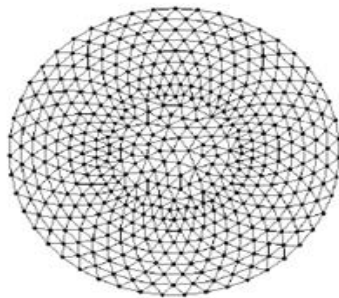


(b) After Optimization of Planar Network

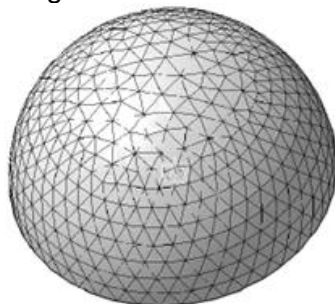


(c) The length of the relative error of cloud

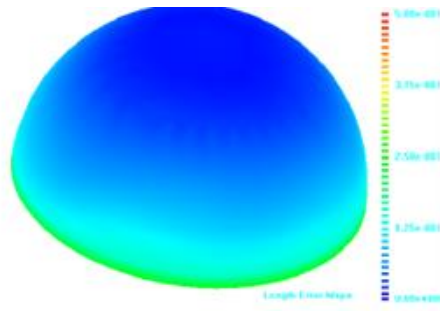
Figure 5. Variational Interpolation Method to Optimize Unicom more Surface Area



(a) Original Surface Grid Model



(b) After Optimization of Planar Network



(c) The Length of the Relative Error of Cloud

Figure 6. Variational Interpolation Method to Optimize Half Spherical Surface

There are continuous vertex sequences for boundaries of two surfaces $P_1 = \{b_1^1, \dots, b_1^n\}$, $P_2 = \{b_2^1, \dots, b_2^n\}$. If some points in these two sequences are coincident, in other words $b_1^i = b_2^j$, these two boundaries could be combined as one with a vertex sequence.

$$P1 \oplus P2 = \{ \dots, b_1^{i-1}, b_2^{j+1}, \dots, b_2^{j-1}, b_1^{i+1}, \dots \} \tag{7}$$

Figure 6 illustrates an example of a developable surface interpolating a smoothly connected boundary. As shown in Figure 7, this synthetic boundary is made up of intersecting boundaries of three cones. The optimal triangularization operation in this paper, as well as a commonly used bridge triangularization optimization, is given by Figure 7(a) and 7(b), respectively. The figure images apparently prove a superior result of methods in this paper. Furthermore, this example also claims limitations of the optimal bridge triangularization in special cases with unusual structures that overcame by the optimal triangularization.



(a) Original surface grid model

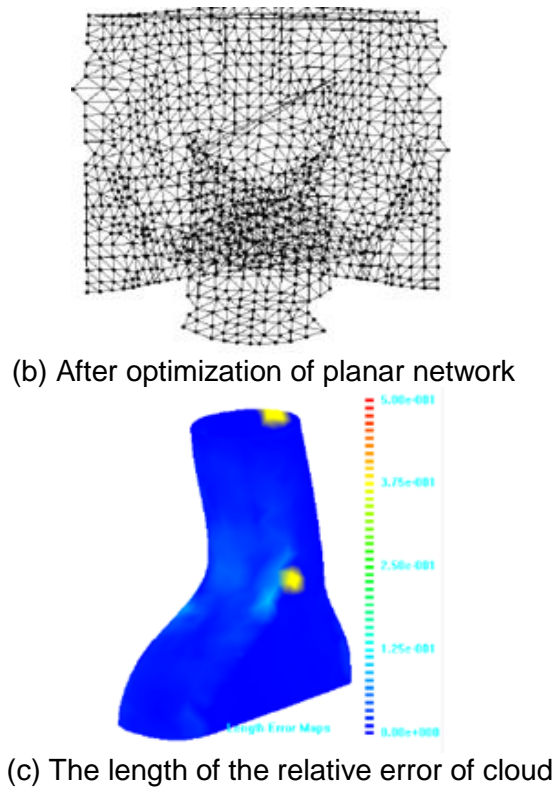
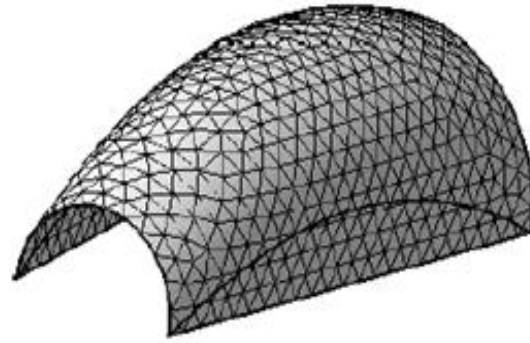


Figure 7. Variational Interpolation Method to Optimize Special Surface

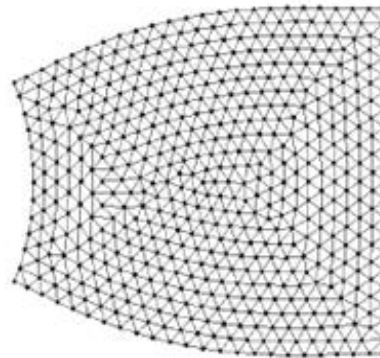
3.3. Quadrangular grid transformations for the optimal developable surface

In industrial field, both quadrangular grids and triangular grids are commonly accepted representations for discrete surfaces. The optimal triangularization could be transformed as follows to reach a quadrangular grid to initially input the optimal PQ grid calculation. Suppose that the optimal triangularization of p is $G = (V, E)$, in which $V \subseteq P$. For each vertex $v \in V$, it is a candidate vertex if the degree $\deg(v) \geq 3$. A side $e \in E$ with two candidate vertex is called a candidate side of e . The dihedral angel of each candidate side ce is presented as $Dih(ce)$. The following describes the process of transforming the optimal triangularization G into a quadrangular grid in detail.

1. Calculate the degree of each vertex of G , and find out all candidate vertexes.
2. Screen out all candidate sides to allocate in a heap h . For each candidate side, give a weight of $Dih(ce)$.
3. If h is not null,
 - 3.1 Find out a candidate side ce with the minimum degree value.
 - 3.2 Refresh h by digging out ce from E .
4. Refresh G , and recalculate all candidate vertexes.
5. For each candidate vertex v ,
 - 5.1 Make 2 copies of $\deg(v)$ for v , and put these copy points in a v -centered interval $[-\varepsilon, \varepsilon]$.
 - 5.2 Partially refresh the other copy of v to force its degree value to be 3.



(a) Original surface grid model



(b) After optimization of planar network



(c) The length of the relative error of cloud

Figure 8. Variational Interpolation Method to Optimize Raised Surface

Figure 8 presents two outcomes of transforming from the optimal triangularization to quadrangular grid. But it also releases that the quadrangular grids generated by methods in this paper may contain few polygons.

3.4. Aesthetic and Architectural Designs

This draft system also provides a convenient modeling tool for developable surface design in the field of aesthetics and architecture. Models of a surface of revolution in Figure 9 consist of 4 and 12 quasi pieces of developable surface, respectively. A student spent about 75 minutes to design all 36 quasi pieces of developable surface of a hemisphere model in Figure 10, and then generated 2304 triangles according to the optimal triangularization.

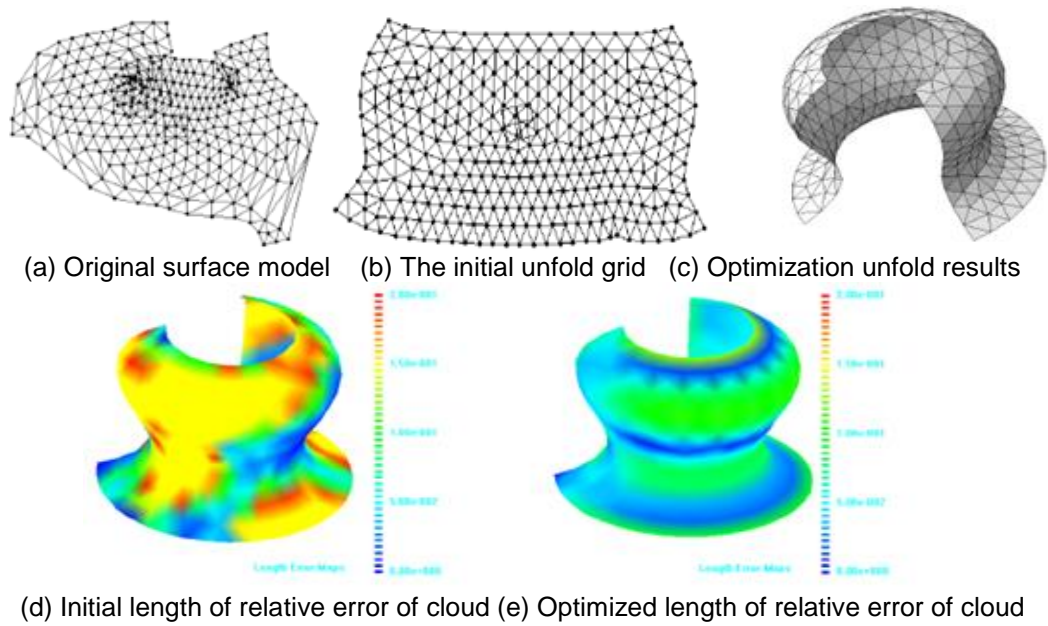


Figure 9. An Optimization Example of Revolution Surface

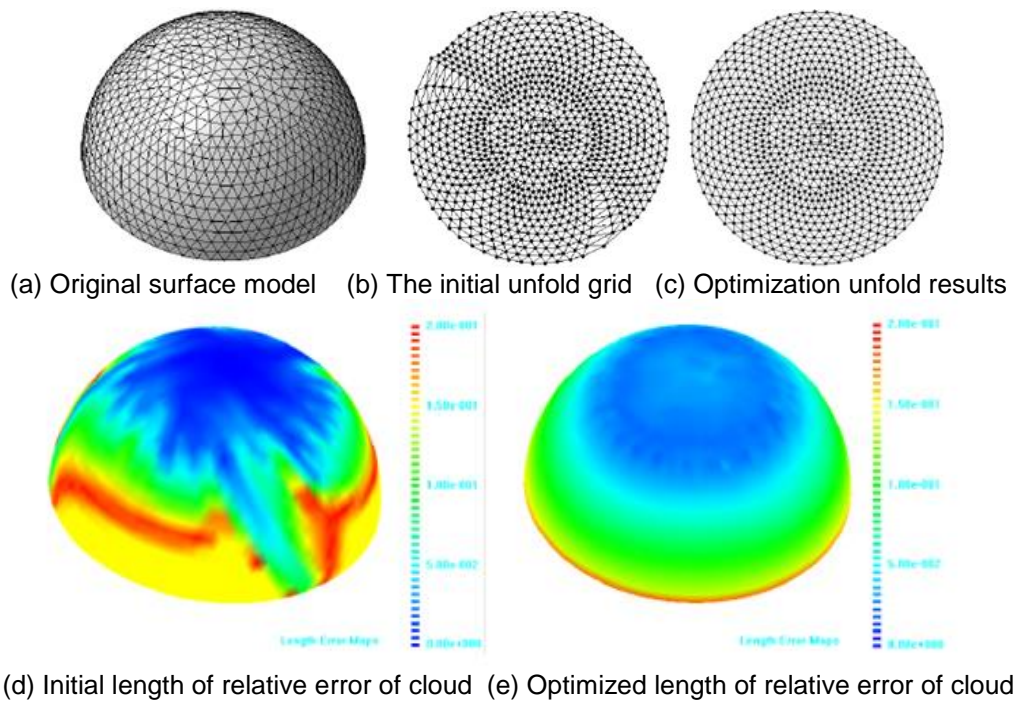


Figure 10. An Optimization Example of Hemisphere Surface

4. Conclusion

This paper introduced a dynamic programming method to generate discrete developable surface for interpolation on certain surface boundaries. In contrast with the previous methods, this algorithm distributed whole process of developable surface design into three stages. First, interactively design the surface boundary conditions in system. Second, interactively modify the curve shapes. Third, generate discrete developable surface automatically. Different from the previous modeling on strip, this interactive draft interface allows users to design surface efficient and effectively. In this study, a lot of cases have verified the resentful outcomes of the algorithm.

Acknowledgment

The authors are grateful to the referee for his/her helpful comments. This work was supported by Educational Commission of Fujian Province, China (Grant No. JA13235) and Natural Science Foundation of Fujian Province, China (Grant No. 2015J01670).

References

- [1] B.K. Hinds, J. McCartney, and G. Woods, Pattern development for 3D surfaces, *Computer-Aided Design*, 23, 8, 583-592, (1991).
- [2] L. Parida, S. P. Mudur, Constraint-satisfying planar development of complex surfaces, *Computer-Aided Design*, 25,4,225-232, (1993).
- [3] P. N. Azariadis, N. A. Aspragathos, Design of plane developments of doubly curved surfaces, *Computer-Aided Design*, 29,10, 675-685, (1997).
- [4] P. N. Azariadis, N. A. Aspragathos, on using planar developments to perform texture mapping on arbitrarily curved surfaces, *Computers&Graphics*, 24,4,539-554, (2000).
- [5] Li, S., Liu, X., Zheng, X., and Lu, H, An Overview of Flattening Methods for Complex Surface, *Journal of Information & Computational Science*, 11, 323-333, (2014).
- [6] T. Shimada, Y. Tada, Approximate transformation of an arbitrary curved surface into a plane using dynamic programming, *Computer-Aided Design*, 23,2,153-159, (1991).
- [7] Feng, Z., Champlaud, H., Sabourin, M., and Morin, S, Optimal blank design based on finite element method for blades of large Francis turbines, *Simulation Modelling Practice and Theory*, 36, 11-21, (2013).
- [8] Bennis, C., Borouchaki, H., Dumont, C., Lerat, O., Poudret, M., and Rainaud, J. F, 3D line-support grid flattening for more accurate geostatistical reservoir population with petrophysical properties, *Engineering with Computers*, 30,3, 403-421, (2014).
- [9] Yu, G., Patrikalakis, N. M., & Maekawa, T, Optimal development of doubly curved surfaces. *Computer Aided Geometric Design*, 17, 6, 545-577, (2000).
- [10] Yang, Y. L., Kim, J., Luo, F., Hu, S. M., and Gu, X, Optimal surface parameterization using inverse curvature map. *IEEE Transactions on Visualization and Computer Graphics*, 14,5, 1054-1066, (2008).
- [11] Liu, X., Li, S., Zheng, X., and Lin, M, Development of a flattening system for sheet metal with free-form surface, *Advances in Mechanical Engineering*, 8,2, 1-12, (2016).

