Interpolating Curved Grids Using Quasi-developable Bi-cubic Gregory Patches

Ming Chen[1], Huakun Yu[1], Kai Tang[2] and Ajay Joneja[3]

[1]Harbin Institute of Technology, hustcm@hotmail.com
[2]The Hong Kong University of Science and Technology, mektang@ust.hk
[3]The Hong Kong University of Science and Technology, joneja@ust.hk

ABSTRACT

The problem of interpolating a quadrilateral free-form network with a quasi-developable patch is investigated in this paper. The boundaries of the quadrilateral are assumed to be degree 3 Bezier curves. The interpolated surface is a bi-cubic Gregory patch, which guarantees G^1 Continuity. All inner control points of the Gregory patch are determined by imposing G^1 continuity constraint and solving an optimization to maximize the surface developability. The optimization model minimizes the value of the integral Gaussian curvature over the entire patch. Examples are provided to illustrate the application of this approach in several industrial applications.

Keywords: developable surfaces, Gregory patch, surface interpolation, Gaussian curvature

1. INTRODUCTION

A surface is developable if it can be flattened onto a plane without distortion, i.e., without any stretching or compression. Such a surface has zero Gaussian curvature everywhere. Because of this characteristic, developable surfaces are widely used in the engineering fields such as ship-building [20] and garment industries [8, 25, 26], and extensive research work has been done in developable surface modeling.

Much work has been done on how to represent a developable surface using polynomial patches. One approach to achieve is to use developable or quasi-developable polynomial patches to interpolate a set of boundary curves [1–3, 5–7, 10, 11] with G^0, G^1 and G^2 continuity. Another approach is to use piecewise linear approximations: the free from surface is tessellated, typically using triangles, and an optimization model is constructed to minimize some form of distortion as the mesh in a 2-parameter space [8, 19, 25, 26]. Since a developable surface can be regarded as a one parameter *family* of straight lines, it can be mapped to one space curve in the dual space. Some researchers [4, 23, 24] convert the modeling of developable surfaces by an equivalent modeling of a curve in dual space. This conversion can guarantee the final surface is developable; unfortunately, the modeling process is not geometrically intuitive, and hence impractical. A practical approach is to sketch the curve-network and then seek a developable surface to interpolate it. Computationally, there may be no solution for an interpolating developable surface; hence some approximation is utilized. In industries where materials such as metals, fabrics or leather are used, this is equivalent to utilizing the available plastic deformation of the materials. In such approximations, one possible objective function defines a surface that minimizes a measure based on the Gaussian curvature.

In this work, a novel algorithm is proposed to interpolate a quadrilateral curve network, denoted as a *curved grid*. Using self-defined cross-derivative functions, a bi-cubic quasi-developable Gregory patch is derived to "tile" the given *curved grid*, with G^1 continuity between neighboring grids along their shared boundary. The use of Gregory patches allows us to handle the vertex enclosure problem [21, 22]; further, we can divide any curve-network with an arbitrary number of boundary curves into a curve ribbon consisting of a series of *curved grids* (see Fig. 1). Another work interpolating a curve-network using Gregory patches can be found in [9], and the curve-network is automatically constructed from a solid which is generated via a set of local modifications on a polyhedron. Different from the work [9], the factor of developability is considered in our work. The measure of developability we use is the integral of the Gaussian curvature over the interpolating surface. After devising an appropriate construction scheme and

Computer-Aided Design & Applications, 12(5), 2015, 564–571, http://dx.doi.org/10.1080/16864360.2015.1014734

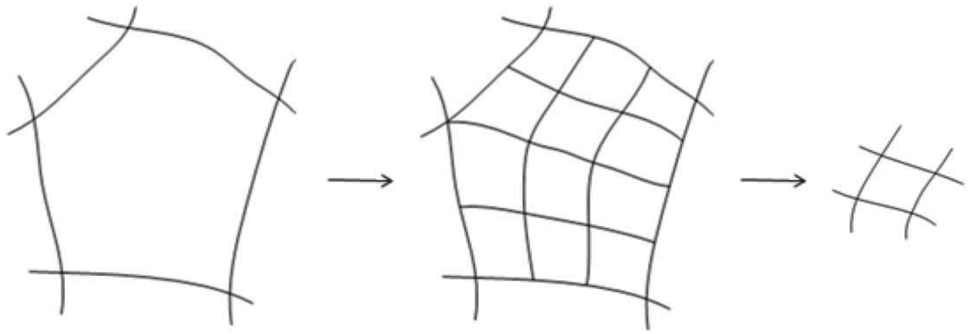

Fig. 1: Dividing a given network into a set of curved grids.

optimizing some free parameters based on the developability [5], a quasi-developable Gregory patch that passes through each *curved grid* is designed. Finally a visually smooth quasi-developable surface can be designed to interpolate the given curve-network.

2. PRELIMINARIES

When interpolating a polynomial curve network with G^1 continuity, one encounters the vertex enclosure problem (discussed in various related past works on surface interpolation, such as [12–14, 18, 28]). Our approach using Gregory patches are used to interpolate a given polynomial curve network. In this paper, bi-cubic Gregory patches are chosen to be as the tilting element, and for ease of computation, we restrict our patches to degree three, which is the minimum required to guarantee G^1 continuity across neighboring patches.

2.1. Bi-cubic Gregory Patch

The parametric form of a bi-cubic Gregory patch is similar to the bi-cubic Bezier surface representation except that the inner control points are related to the parameters *u* and *v*.

$$G(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} B_i^3(u)\, B_j^3(v) P_{ij}(u,v) \qquad (0 \le u, v \le 1) \tag{1}$$

Where $B_i^3(u)$and $B_j^3(v)$ are the Bernstein polynomial of degree 3 and P_{ij} are control points. From Fig. 2, we need to determine 20 control points while there are only 16 control points are needed. Actually, the eight inner points (blue in Fig. 2) are blended into four. In other words, we have $P_{ij}(u,v) = P_{ij0} = P_{ij1}$, except for the inner control points, i.e., P_{11}, P_{21},P_{12} and P_{22} , which can be evaluated as:

$$P_{11}(u,v) = \frac{uP_{110} + vP_{111}}{u+v} \tag{2}$$

$$P_{21}(u,v) = \frac{(1-u)P_{210} + vP_{211}}{1-u+v} \tag{3}$$

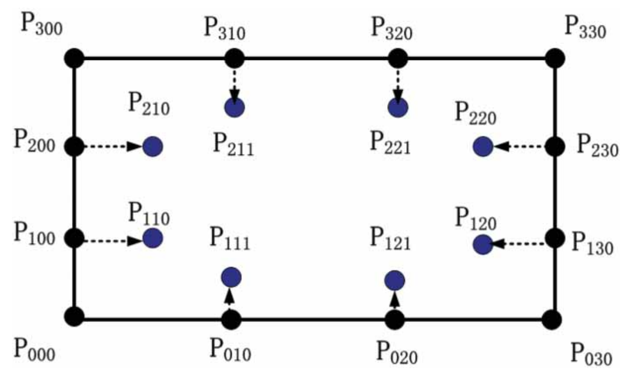

Fig. 2: The control points of bi-cubic Gregory patch.

$$P_{12}(u,v) = \frac{uP_{120} + (1-v)P_{121}}{u+1-v} \tag{4}$$

$$P_{22}(u,v) = \frac{(1-u)P_{220} + (1-v)P_{211}}{2-u-v} \tag{5}$$

The first and second cross-derivatives along the boundary of the Gregory patch are equal to that of the corresponding Bezier patch with the same boundary curves. More explicitly, the first cross-derivatives along the boundary $G\,(u,\,0)$, can be evaluated as:

$$\frac{\partial G}{\partial v}(u,0) = 3\sum_{i=0}^{3} B_i^3(u)(P_{1i} - P_{0i}) \tag{6}$$

Using the corresponding relationship between $P_{ij}(u,v)$ in Eq.(1) and P_{ijk} in Fig. 2 and the Eq.(6) can be expanded as Eq.(7) below:

$$\begin{aligned}\frac{\partial G}{\partial v}(u,0) = 3[&B_0^3(u)(P_{100} - P_{000}) + B_1^3(u)(P_{111} - P_{010}) \\ &+ B_2^3(u)(P_{121} - P_{020}) + B_3^3(u)(P_{130} - P_{030})]\end{aligned} \tag{7}$$

All control points of Gregory patch coincide with that of the given boundary curves. If the degree of a boundary curve is less than 3, it is assumed to be 3 with the coefficients of higher order terms set to 0. If the twist at a corner has different values, e.g.,

$\frac{\partial^2 G}{\partial u \partial v}(0,0) \neq \frac{\partial^2 G}{\partial v \partial u}(0,0)$, the resulting system is singular. This can be resolved in different ways [13]. In our approach, we force the projection components of the twist partial values on the given normal vector N at corner to be equal. All eight internal control points are unknown and need to be determined by G^1 continuity and developability condition.

2.2. Continuity Between Two Adjacent Gregory Patches

In order to better understand G^1 geometric constraint, one can imagine a virtual patch exists across the shared boundary of two adjacent Gregory patches. If these two patches have the same tangent plane with the virtual patch respectively, they are joined in a G^1 continuity [16, 28]. In Fig. 3, $R(u)$ is the intersection line of these two neighboring patches denoted as G and B. The virtual patch across $R(u)$ is denoted as V, which is G^1 connected with both A and B. The polynomial D (u) is the first cross-derivatives of patch V along $R(u)$. $G_v(u,0)$ and $A_w(u,0)$ are first cross-derivatives of patch G and patch B along $R(u)$,respectively. If patch V joins both patch G and B in a G^1 continuous manner, the following equations, i.e., Eq.(8) and Eq.(9) should hold as below:

$$G_v(u,0) = a(u)D(u) + b(u)\dot{R}(u) \tag{8}$$

$$B_W(u,0) = c(u)D(u) + d(u)\dot{R}(u) \tag{9}$$

where $a(u)$, $b(u)$, $c(u)$ and $d(u)$ are polynomials. Since $D(u)$ independently interpolates the derivatives at two end vertices of $R(u)$, it should be of degree 3; as the degrees of left sides in Eq.(8) and Eq.(9) are both 3, therefore $a(u)$, $c(u)$ are constant, and $b(u)$ and $d(u)$ are either linear or constant, since $\dot{R}(u)$ is of degree 2. In our work, we let $b(u)$ and $d(u)$ to linear, thereby getting two free parameters for optimizing the developability. If Eqs. (8)-(9) hold, patch A and patch B are also G^1 continuous.

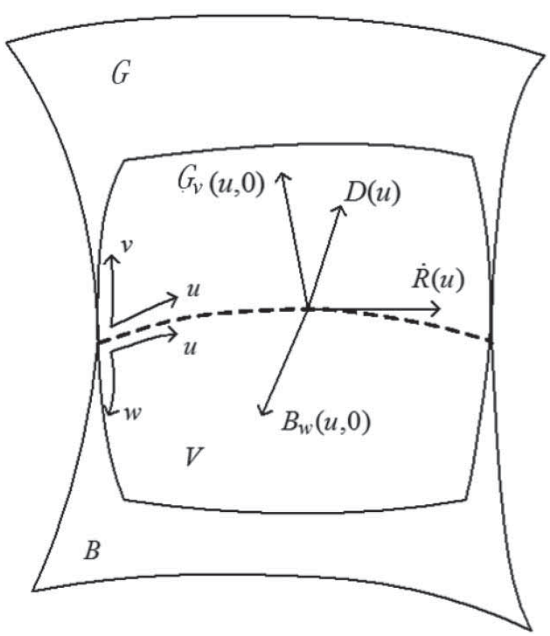

Fig. 3: The virtual patch between the two adjacent patches.

3. METHODOLOGY

As mentioned before, we need to determine eight unknown inner control points based on G^1 geometric condition, i.e., Eq.(8) and Eq.(9). Eq.(8) and Eq.(9) independently specify the relationship between unknown control points and G^1 condition of two neighboring patches G and B, respectively. Thus if we can determine the inner control points patch G using Eq.(8), the control points for patch B can be determined similarly.

3.1. Determine *D*(u)

The mixed partial derivative of $G(u, v)$, $\dot{D}(u)$, is one parameter used in calculating the Gaussian curvature of boundary of G; therefore we cannot set $\dot{D}(u)$to zero as a constraint in our optimizing model. Further, since the degree of the left side of Eq.(8) is 3, we let the degree of $D(u)$ as 3, i.e., $D(u) = \sum_{i=0}^{3} B_i^3(u)D_i$. Thus, we have $D(0) = D_0$, $D(1) = D_3$, $\dot{D}(0) = 3(D_1 - D_0)$ and $\dot{D}(1) = 3(D_3 - D_2)$. If we can determine $D(0)$, $D(1)$ and their first derivative values, i.e., $\dot{D}(0)$ and $\dot{D}(1)$, $D(u)$ will be fully determined. We first consider $\dot{D}(0)$.

In Fig. 4, R_1 and R_2 are the two boundary curves of patch G and $\dot{D}$is the twist, i.e., the mixed partial derivative of $G(u, v)$ at the corner in the direction u. X and Y are the principal vectors of G, and $\vec{X}$and $\vec{Y}$are their corresponding unit vectors, which define two orthogonal direction and N_1 is the unit normal vector at the corner, which is calculated as$N_1 = \frac{\dot{R}_1(0)\times\dot{R}_2(0)}{||\dot{R}_1(0)\times\dot{R}_2(0)||}$. The Gaussian curvature is given by:

$$K = \frac{LN - M^2}{EG - F^2} \tag{10}$$

Where, E, F, G are the coefficients of the first fundamental form and L, M, N are those of the second fundamental form. After direct evaluation by Eq.(10), the Gaussian curvature of the corner P_{000} can be evaluated as Eq. (11).

$$K(0) = \frac{\langle \ddot{R}_1(0), N_1\rangle\langle \ddot{R}_2(0), N_1\rangle - T^2}{\dot{R}_1^2(0)\dot{R}_2^2(0) - \langle \dot{R}_1(0), \dot{R}_2(0)\rangle^2} \tag{11}$$

Where $<\ ,\ >$ denotes the vector dot product, $T = \langle \dot{D}(0), N_1\rangle$, which equals to the twist of $G(0,0)$. Since the interpolating patch is preferred to be developable, $K(0)$ is preferred to be as close to

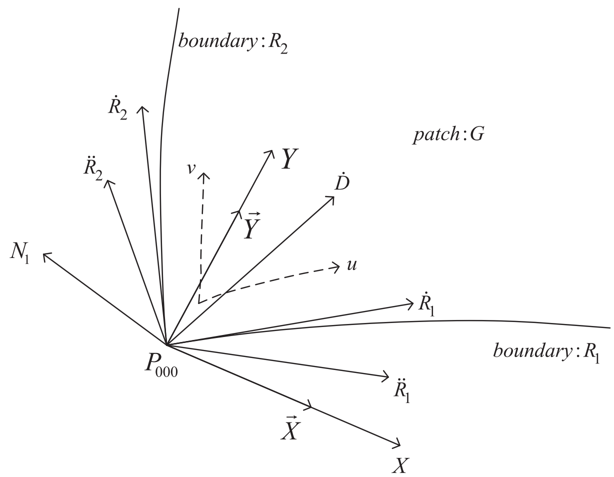

Fig. 4: The involved vectors at the corner P_{000} of patch G.

zero as possible. $\langle \ddot{R}_1(0), N_1\rangle\langle \ddot{R}_2(0), N_1\rangle > 0$, we let $T = \sqrt{\langle \ddot{R}_1(0), N_1\rangle\langle \ddot{R}_2(0), N_1\rangle}$, otherwise let $T = 0$.

So far only the component of $\dot{D}(0)$ along the direction of N_1, i.e., $\langle \dot{D}(0), N_1\rangle$ is evaluated. We need to calculate the component projected onto the tangent plane. Thus, we first evaluate two principal directions $\vec{X}$and $\vec{Y}$ (where $\vec{X}\cdot\vec{Y} = 0$), and construct a frame, to write $\dot{D}(0)$ in terms of N_1, $\vec{X}$ and $\vec{Y}$.

We can easily calculate the mean curvature $H(0)$ as below:

$$H(0) = \frac{1}{2}\frac{\langle \ddot{R}_2(0), N_1\rangle \dot{R}_1^2(0) - 2T\langle \dot{R}_1(0), \dot{R}_2(0)\rangle + \langle \ddot{R}_1(0), N_1\rangle \dot{R}_2^2(0)}{\dot{R}_1^2(0)\dot{R}_2^2(0) - \langle \dot{R}_1(0), \dot{R}_2(0)\rangle^2} \tag{12}$$

Denote two principal normal curvature as $K_{\max}$ and $K_{\min}$ and their corresponding vectors as K and L, respectively. Based on the definition of Gaussian curvature and Mean curvature, we obtain:

$$\begin{cases} K(0) = K_{\max}K_{\min} \\ H(0) = \frac{K_{\max}+K_{\min}}{2} \end{cases} \tag{13}$$

$K_{\max}$, $K_{\min}$ and the principal direction X and Y can be calculated from Eqs.(11) –(13) as:

$$\begin{cases} X = (T - K_{\max}F)\dot{R}_1(0) - (\langle \ddot{R}_1(0), N_1\rangle \\ \qquad -K_{\max}\langle \dot{R}_1(0), \dot{R}_2(0)\rangle)\dot{R}_2(0) \\ Y = (T - K_{\min}F)\dot{R}_1(0) - (\langle \ddot{R}_1(0), N_1\rangle \\ \qquad -K_{\min}\langle \dot{R}_1(0), \dot{R}_2(0)\rangle)\dot{R}_2(0) \end{cases} \tag{14}$$

Note that $K_{\min}$ may be not zero when curves R_1 and R_2 do not bend toward the same direction. The corresponding unit vector $\vec{X}$and $\vec{Y}$ can be easily obtained by normalizing the vectors X and Y; $\vec{X}$, $\vec{Y}$ and N_1 define a local frame, denoted by $\vec{X}\vec{Y}N_1$ with the origin at point P_{000}. The component of $\dot{D}_0$ along the direction N_1 is equal to $T = \langle \dot{D}(0), N_1\rangle$, and the other two component values along the $\vec{X}$ and $\vec{Y}$ axes are denoted as x and y, respectively; both of these must be specified. Since $\dot{D}_0 = 3(D_1 - D_0)$, we obtain:

$$D_1 = D_0 + \frac{\langle \dot{D}_0, N_1\rangle}{3}N_1 + x\vec{X} + y\vec{Y} \tag{15}$$

The optimum values of x and y can be evaluated by minimizing the value of $[(D_1 - D_0) - \frac{1}{3}(D_3 - D_0)]^2$ such that $D(u)$ can be more smoothly distributed, which yields:

$$\begin{cases} D_1 = D_0 + \frac{\langle \dot{D}_0, N_1\rangle}{3}N_1 + \left\langle \frac{D_3 - D_0}{3}, \vec{X}\right\rangle \vec{X} \\ \qquad + \left\langle \frac{D_3 - D_0}{3}, \vec{Y}\right\rangle \vec{Y} \\ D_2 = D_3 - \frac{\langle \dot{D}_3, N_2\rangle}{3}N_2 + \left\langle \frac{D_3 - D_0}{3}, \vec{X}_2\right\rangle \vec{X}_2 \\ \qquad + \left\langle \frac{D_3 - D_0}{3}, \vec{Y}_2\right\rangle \vec{Y}_2 \end{cases} \tag{16}$$

Where $\vec{X}_2\vec{Y}_2N_2$ is a local frame at the corner P_{030}. From Eq. (16), we can see that D_1 and D_2 are related to D_0 and D_3. We now evaluate D_0; D_3 can be similarly determined. Let $u = 0$, 1, and rewrite Eq. (8) in terms of the derivative of R_1 and R_2, we obtain:

$$\begin{cases} \dot{R}_2(0) = a(0)D_0 + b(0)\dot{R}_1(0) \\ \dot{R}_2(1) = a(1)D_3 + b(1)\dot{R}_1(1) \end{cases} \tag{17}$$

As the left side of Eq.(8) and $D(u)$ are both of degree 3, $a(0) = a(1) = a$, and without loss of generality, we let $a = 1$. D_0 , $\dot{R}_2(0)$ and $\dot{R}_1(0)$should be all on the tangent plane spanned by unit vector $\vec{X}$and $\vec{Y}$. We represent them in terms of $\vec{X}$and$\vec{Y}$ as below:

$$\begin{cases} D_0 = x_0\vec{X} + y_0\vec{Y} \\ \dot{R}_1(0) = m_1\vec{X} + n_1\vec{Y} \\ \dot{R}_2(0) = m_2\vec{X} + n_2\vec{Y} \end{cases} \tag{18}$$

Substituting Eq. (17) into Eq. (18), we obtain:

$$\begin{cases} x_0 = (b(0)m_1 - m_2) \\ y_0 = (b(0)n_1 - n_2) \end{cases} \tag{19}$$

Thus D_1 is related to the parameter b (0) and denoted it as b_0, similarly D_2 is related to the parameter $b(1)$ denoted as b_1. So far we have determined D_i as a bivariate function in terms of b_0 and b_1 expressed as:

$$D(u) = f_0(b_0)B_0^3(u) + f_1(b_0, b_1)B_1^3(u) + f_2(b_0, b_1)B_2^3(u) + f_3(b_1)B_3^3(u) \tag{20}$$

where f_i is linear function with respect to b_0 and b_1.

3.2. Determining the Eight Interior Control Points

Substituting the value of $D(u)$ in Eq. (20) back to Eq. (8) and rearranging gives:

$$3\sum_{i=0}^{2} B_i^3(u)(P_{1i} - P_{0i}) = \sum_{i=0}^{3} f_i B_i^3(u) + 3(b_0 + u(b_1 - b_0)) \times \sum_{i=0}^{2} B_i^2(u)(Q_{i+1} - Q_i) \qquad (21)$$

where Q_i are control points of the curve R_1. Using Eq. (7) and Eq. (21), we can represent two unknown inner control points, P_{111} and P_{121}, as a linear function of b_0 and b_1, i.e., $P_{111}(b_0, b_1)$ and $P_{121}(b_0, b_1)$. Similarly, the remaining six unknown control points, i.e., $P_{110}, P_{210}, P_{211}, P_{221}, P_{220}, P_{120}$, can also be represented as linear functions of three pairs of parameters, (r_0, r_1), (s_0, s_1) and (t_0, t_1).

We wish to achieve a (near)-developable surface; to do so, we shall adopt an objective function that minimizes the Gaussian curvature aggregated over the surface. We denote this criterion as $g(b_0, b_1, r_0, r_1, s_0, s_1, t_0, t_1)$ and define it as:

$$g(b_0, b_1, r_0, r_1, s_0, s_1, t_0, t_1) = \iint_G |K| du dv \qquad (22)$$

where K is Gaussian curvature at the point $G(u, v)$, represented parametrically via Eq. (10). If the curved grid is relatively small and smooth, then we may assume that the sign of the Gaussian curvature does not change across the entire surface. In this case, the computation can be simplified by employing the Gauss-Bonnet theorem:

$$\iint_G |K| du dv = |2\pi - \sum_{i=1}^{4} \oint k_g(u) dL - \sum_{i=0}^{4} \alpha_i| \qquad (23)$$

Where k_g is geodesic curvature along given boundary curves, and $\alpha_i$denotes the external angles between the tangent vectors at four corners. We can solve the values of $(b_0, b_1, r_0, r_1, s_0, s_1, t_0, t_1)$ that minimize our objective function, g(.), by using the appropriate numerical approach; The solution is then used to calculate the interior control points using Eq. (21). We note that all initial inner control points are calculated using the method in [14], and the initial values of $(b_0, b_1, r_0, r_1, s_0, s_1, t_0, t_1)$ are correspondingly calculated easily by Eq. (21). In our case, we employ BFGS-quasi-Newton's method [15] which converges to a stationary point, i.e., a local minimum in general. Better solutions, possibly closer to the global optimum may be obtained by employing this within some randomized search algorithm (e.g. simulated annealing) [17, 27]. In our algorithm, the gradient vector is numerically calculated and the Hessian matrix is efficiently approximated. The ARPREC Library is used to solve this optimization problem due to its tested robustness against numerical instability.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed algorithm has been implemented using C++ and tested on a PC with Intel i5-760 3G Hz, 2GB DDR2. To verify the algorithm, the benchmark is the method in [14], which constructs a G^1 bi-cubic Gregory patch without considering the developability constraint. In [14], G^1 continuity constraints are given and inner control points are initially estimated using a

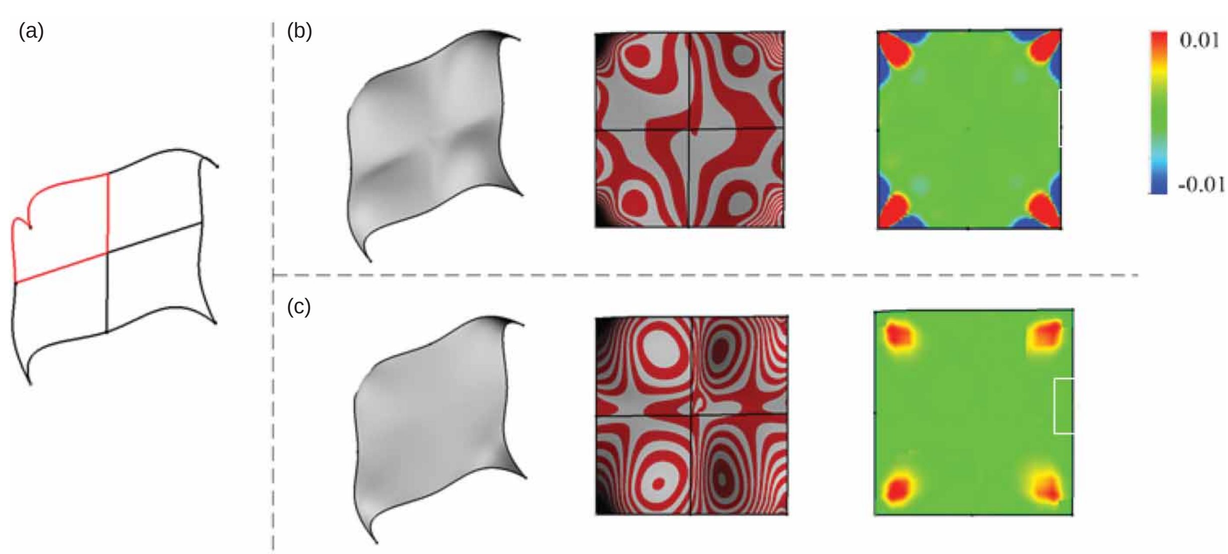

Fig. 5: (a) A curve network with four curved grids, each of which consists four planar curves of degree 3; (b) the result by [14];(c) the result by the proposed algorithm.

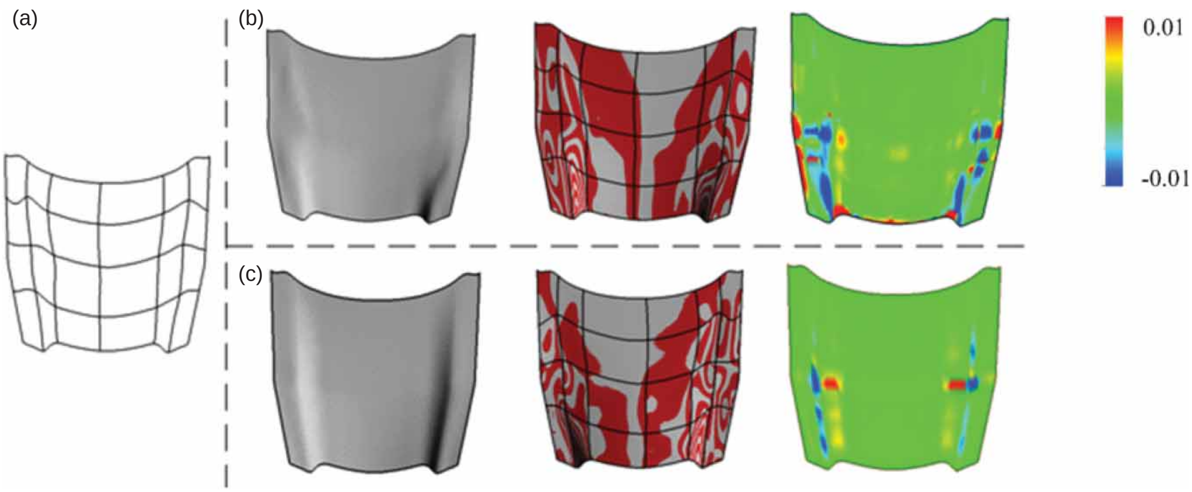

Fig. 6: (a) A curve network with 24 curved grids, each of which consists four planar curves of degree 3; (b) the result by [14]; (c) the result by the proposed algorithm.

heuristic scheme. Next, these control points are constructed in a manner which satisfies G^1 conditions; finally the solution is optimized by minimizing the difference between the final positions and their corresponding initial estimated positions. This method only focuses on G^1 continuity and does not consider how to devise the first order and second order derivative fields along boundary curves, which will in turn used to determine all inner control points. In our paper, the derivatives along the boundary curves are computed to minimize the Gaussian curvatures at four corner points; so in principle, our method will improves on [14] if developability of the interpolating surface is desirable. A visual comparison of some examples constructed by out method and by using the approach in [14] is presented via Zebra maps and Gaussian maps in Figure 5. In the Gaussian map, blue color indicates (nearly) zero Gaussian curvature, while green/red regions have respectively negative/positive Gaussian curvature.

The first example is a simple one with four curved *grids* (see Fig. 5(a)), each of which consists of four 3-degree-planar curves and located symmetrically. The results by [14] and our methods are shown in Fig. 5(b) and Fig. 5(c), respectively. Their corresponding shaded view, zebra-pattern rendering and the distribution map of Gaussian curvature are shown from the left to the right in Fig. 5(b) and Fig 5(c). From the reflection model of Fig. 5(c), we can see the resulting patch is G^1 continuous and the distribution of the Gaussian curvature is much better (from a point of view of developability) than that in Fig. 5(b). We also see that the Gaussian curvature at all the corners is zero, which means the surface is locally planar in the neighborhood of these points.

The second example is a car hood formed using sheet-metal. The shape is preferred to be as developable as possible so that the internal stresses are minimized. In this example, ribbon curves are constructed first, making 24 curved grids as shown in Fig. 6(a). The results by using the benchmark and by our algorithm are shown respectively in Fig. 6(b) and Fig. 6(c). Again, these rendered views clearly indicate the effectiveness of our method in producing a more developable surface.

5. CONCLUSIONS

In this paper, a novel algorithm is proposed to construct a degree-3 Gregory patch interpolating a curved *grid* consisting of four Bezier curves of degree 3. In this algorithm, the first cross derivatives along boundary curves are optimally specified by imposing a developability constraint and all inner unknown control points are evaluated by ensuring G^1 continuity. The developability constraint used in our implementation minimizes the integral of the magnitude of the Gaussian curvature over the whole patch; other similar criteria can be easily adopted into our approach. Test examples showed that the proposed algorithm can be applied to many industries in which developability is desirable. Our approach can be further improved in future work, to address some limitations, which include the following.

The construction of Gregory patch only guarantees G^1 continuity, but there are many applications where G^2 or even higher order continuity is desirable. Further, the surface is constructed by optimizing the integral Gaussian curvature over the entire Gregory patch. It is not clear that this objective is a reasonable metric in all applications, since even a global optimum under this criterion may have some local regions with excessively high Gaussian curvature. One way to address this may be to use a constraint that limits the maximum local Gaussian curvature to within a preset value.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China (Grant No. 61100088), the Natural Science Foundation of Guangdong Province (Grant No. S2013010013133, No. S2011040000145) and the Fundamental Research Projects of Shenzhen (Grant No. JC201105160526A).

REFERENCES

[1] Aumann, G.: A simple algorithm for designing developable Bézier surfaces, Computer Aided Geometric Design, 20(8), 2003, 601–619. http://dx.doi.org/10.1016/j.cagd.2003.07.001

[2] Aumann, G.: Degree elevation and developable Bézier surfaces, Computer Aided Geometric Design, 21(7), 2004, 661–670. http://dx.doi.org/10.1016/j.cagd.2004.04.007

[3] Aumann, G.: Interpolation with developable Bézier patches, Computer Aided Geometric Design, 8(5), 1991, 409–420. http://dx.doi.org/10.1016/0167-8396(91)90014-3

[4] Bodduluri, R. M. C.; Ravani, B.: Design of developable surfaces using duality between plane and point geometries, Computer-Aided Design, 25(10), 1993, 621–632. http://dx.doi.org/10.1016/0010-4485(93)90017-I

[5] Chen, M.; Tang, K.: G^2 quasi-developable Bezier surface interpolation of two space curves, Computer Aided Design, 45(11), 2013, 1365–1377. http://dx.doi.org/10.1016/j.cad.2013.06.009

[6] Chen, M.; Tang, K.: Quasi-developable surface modeling of contours with curved triangular patches, Computers & Graphics, 37(7), 2013, 851–861. http://dx.doi.org/10.1016/j.cag.2013.05.002

[7] Chen, M.; Tang, K.; Joneja, A.: Design of Developable Interpolating Strips, Computer-Aided Design and Applications, 8(4), 2011, 557–570. http://dx.doi.org/10.3722/cadaps.2011.557-570

[8] Chen, M.; Tang, K.: A fully geometric approach for developable cloth deformation, The Visual Computer, 26(6–8), 2010, 853–863. http://dx.doi.org/10.1007/s00371-010-0467-5

[9] Chiyokura, H.; Kimura, F.: Design of solids with free-form surfaces, ACM SIGGRAPH Computer Graphics, 17(3), 1983, 289–298. http://dx.doi.org/10.1145/964967.801160

[10] Chu, C. H.; Séquin, C. H.: Developable Bézier patches: properties and design, Computer-Aided Design, 34(7), 2002, 511–527. http://dx.doi.org/10.1016/S0010-4485(01)00122-1

[11] Chu, C. H.; Wang, C. C. L; Tsai, C. R.: Computer aided geometric design of strip using developable Bézier patches, Computers in Industry, 59(6), 2008, 601–611. http://dx.doi.org/10.1016/j.compind.2008.03.001

[12] Du, W. H.; JM Schmitt, F.: On the G^1 continuity of piecewise Bézier surfaces: a review with new results, Computer-Aided Design, 22(9), 1990, 556–573. http://dx.doi.org/10.1016/0010-4485(90)90041-A

[13] Farin, G.: A construction for visual C^1 continuity of polynomial surface patches, Computer Graphics and Image Processing, 20(3), 1982, 272–282. http://dx.doi.org/10.1016/0146-664X(82)90085-5

[14] Farin, G.; Hansford, D.: Agnostic G^1 Gregory surfaces, Graphical Models, 74(6), 2012, 346–350. http://dx.doi.org/10.1016/j.gmod.2012.05.004

[15] Fletcher, R.: Practical methods of optimization (2nd ed.), New York, 1987.

[16] Hermann, T.: G^2 interpolation of free form curve networks by biquintic Gregory patches, Computer Aided Geometric Design, 13(9), 1996, 873–893. http://dx.doi.org/10.1016/S0167-8396(96)00013-1

[17] Kleinberg, J.; Tardos, E.: Algorithm Design, Addison-Wesley, 2005.

[18] Liu, D.; Hoschek, J.: GC^1 continuity conditions between adjacent rectangular and triangular Bézier surface patches, Computer-Aided Design, 21(4), 1989, 194–200. http://dx.doi.org/10.1016/0010-4485(89)90044-4

[19] Liu, Y. J.; Tang, K.; Gong, W. Y.; et al: Industrial design using interpolatory discrete developable surfaces, Computer-Aided Design, 43(9), 2011, 1089–1098. http://dx.doi.org/10.1016/j.cad.2011.06.001

[20] Pérez, F.; Suárez, J.A.: Quasi-developable-spline surfaces in ship hull design, Computer-Aided Design, 39(10), 2007, 853–862. http://dx.doi.org/10.1016/j.cad.2007.04.004

[21] Peters, J.: Joining smooth patches around a vertex to form a C^k surface, Computer Aided Geometric Design, 9(5), 1992, 387–411. http://dx.doi.org/10.1016/0167-8396(92)90032-K

[22] Peters, J.: Smooth interpolation of a mesh of curves, Constructive Approximation, 7(1), 1991, 221–246. http://dx.doi.org/10.1007/BF01888155

[23] Pottmann, H.; Farin, G.: Developable rational Bezier and B-spline surfaces, Computer Aided Geometric Design, 12(5), 1995, 513–531. http://dx.doi.org/10.1016/0167-8396(94)00031-M

[24] Pottmann, H.; Wallner, J.: Approximation algorithms for developable surfaces, Computer Aided Geometric Design, 16(6), 1999, 539–556. http://dx.doi.org/10.1016/S0167-8396(99)00012-6

[25] Tang, K.; Chen, M.: Quasi-developable mesh surface interpolation via mesh deformation,

IEEE Transactions on Visualization and Computer Graphics, 15(3), 2009, 518–528. http://dx.doi.org/10.1109/TVCG.2008.192

[26] Wang, C.C.L.; Tang, K.: Achieving developability of a polygonal surface by minimum deformation: a study of global and local optimization approaches, The Visual Computer, 20(8–9), 2004, 521–539. http://dx.doi.org/10.1007/s00371-004-0256-0

[27] Whitley, D. A.: genetic algorithm tutorial, Statistics and computing, 4(2), 1994, 65–85. http://dx.doi.org/10.1007/BF00175354

[28] Ye, X.; Liang, Y.; Nowacki, H.: Geometric continuity between adjacent Bézier patches and their constructions, Computer aided Geometric design, 13(6), 1996, 521–548. http://dx.doi.org/10.1016/0167-8396(95)00043-7