Developable Mesh Surface Approximation by Normal-Guided Deformation

Long Zeng
Mechanical Department
Hong Kong University
of Science and Technology
Hong Kong, PRC
Email: zelog@ust.hk

Ming Chen
Mechanical Department
Hong Kong University
of Science and Technology
Hong Kong, PRC
Email: mecm@ust.hk

Matthew Ming-Fai Yuen
Mechanical Department
Hong Kong University of
Science and Technology
Hong Kong, PRC
Email: meymf@ust.hk

***Abstract*—This paper presents a new algorithm to improve the developability (i.e., make Gaussian Curvature close to zero) of a mesh surface by normal guided deformation. It takes advantage of discrete normal image to convert originally highly non-linear problem into a linear problem which can be solved efficiently. Also, the mesh surface can be deformed with position constraints or within user-preset tolerance. The initial experimental tests have shown some promising aspects in improving mesh surface's developability.**

I. Introduction

Developable surface is a special kind of ruled surface with zero Gaussian curvature everywhere. It has wide application in industries such as sheets metal, toys and garments, etc.

In differential geometry [4], the normal map (Gauss map) maps a regular surface M's normal vector to points on a unit sphere S^2. A point on sphere S^2 is called the normal image if it corresponds to some points on surface M. A regular surface whose normal image is a degenerated spherical curve is called developable surface. Conversely, if a regular surface's normal image is a spherical curve, it is a developable surface. This property of normal map can be used to measure a surface developability in surface normal image space (on unit sphere surface S^2), the less the normal image of a surface deviates from an ideal curve, the more developable it is. This deviation is easy to measure comparing to traditional measurement based on integral of the absolute Gaussian curvature over the entire surface. Thus, we can convert originally highly non-linear problem into a linear problem (The ideal curve is a spherical curve. It will be explained in detail in the following section).

Based on this understanding, we present a novel algorithm to efficiently improve a given mesh surface developability by iteratively making its normal image converged to its ideal curve. In order to enable our algorithm to deal with more complicated input mesh, we borrow curvature based mesh segmentation [8] as a pre-processing step. Our algorithm framework consists of three stages as illustrated in Fig.1.

Our new method deforms a mesh surface under the guide of its ideal curve. The highly non-linear Gaussian curvature optimization is converted into a linear one, which is solved efficiently and stably.

II. Previous Work

There are a number of research papers about developable surface design. With different inputs, different objectives are set. First, given a set of boundary curves as constraints, a parametric [2], [5] or [6], [13] discrete developable surface is found to interpolate them. Second, given an arbitrary mesh surface, the objective is to improve its developability satisfying some criteria. This is considered as an optimization problem [12], [14], [11]. Besides, some researchers divide the input mesh into a set of elementary primitives, such as strips [9], [1], or approximated developable segments [8].

Our algorithm also defines the problem as an optimized deformation process. However, our deformation takes advantage of discrete normal map information. This is different with Tang, et al [11], which adopted handle-based shape editing techniques to transform an initial developable mesh surface to a final mesh surface that interpolates a set given anchor points or curves. The paper that closely related to our idea is Decaudin [3], who generated approximated developable surfaces by finding the best-fitting developable surface for each triangle locally. However, it seems they had not relized this could be done globally and in a mesh surface's normal image space. A spherical curve which globally fits them well can be found first. Then this fitting-curve, called ideal curve, is used to guide the whole mesh deformation to make them converge to it.

III. Objective normal vectors

In this paper, we focus on the method to improve a mesh surface's developability based on discrete normal image (i.e., each normal belongs to a triangle instead of a point).

First, to facilitate the ideal curve computation, the normal image is mapped to the sphere parametric domain. Given a mesh patch, as Fig.2(a), we first obtain its normal image (red points in Fig.2(b)) by Gauss map, denoted as V_s. Then the normal image is mapped to parametric domain, as the unordered red point cloud in Fig.2(d), denoted by V_{s2D}.

Second, a curve is found that fits the unorder point cloud V_{s2D} well. Since the ideal curve is used to guide the mesh deformation, it should pass through those skeleton points of V_{s2D} to minimize the deformation afford, i.e., mesh shape

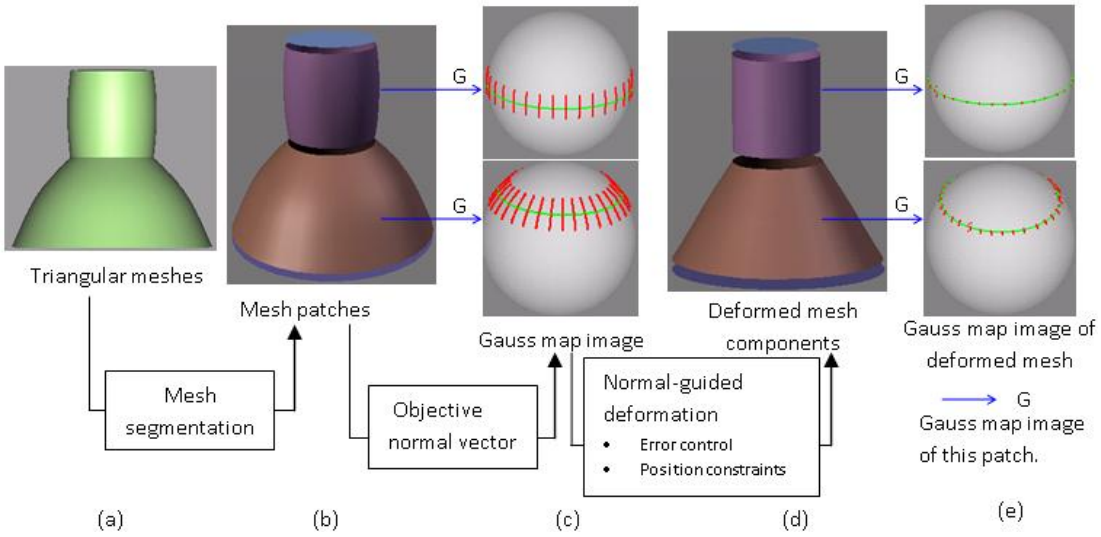

Fig. 1. Algorithm framework. (a→b) the original mesh is segmented into patches according to curvature distribution; (b→c) obtain patch's normal image (red points), fit by an ideal curve (green curve), compute the target points on it corresponding to those red points; (c→d) then each patch deformed in line with the new normal direction, under user-defined constraint and tolerance control.

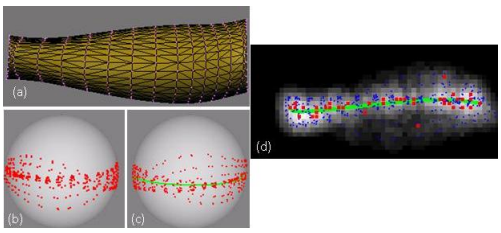

Fig. 2. Spherical curve fit. (a) origin mesh model; (b) normal image; (c) fitting curve (green); (d) digital image in parametric plane.

change. To achieve this objective, we modified Goshtasby et al [7]'s method in several places:

- Add effect region control
- Each grid value is a function of its point number contained in this grid. The more points a grid contains, the larger of the influence value to its surround grids.

These two modifications are expressed by $sign(x)$ and $r(p)$ in Eq.1

$$g(x,y) = \sum_{i=1}^{N} sign(dist(p,p_i))[(x-x_i)^2 + (y-y_i)^2 + r(p)^2]^{1/2} \quad (1)$$

Where $dist(p,p_i)$ is a distance function; $sign(x)$ is a sign function, $sign(x)=1$, if x is less than a threshold, otherwise, $sign(x)=0$; thus, $sign(dist(p,p_i))$ means a reference grid has no influence to those grids whose distance is greater than the given threshold;$\#(x)$ is the number of set x; $r(p) = k \cdot \#(containedpoints)/\#(totalpoints)$, k is a scale factor.

The radial effect field is a digit image, as Fig.2(d). With each grid itself as center, its influence is spreaded radically in the valid region according to Eq.1. Then, based on local maximum and image gradient direction, those skeleton points, the blue points in Fig.2(d), can be found.

Finally, the objective normal vectors are computed. A B-spline curve c_{2D} is used to fit those skeleton points by least square method, the green curve in Fig.2(d). c_{2D} is mapped back to the unit sphere surface, obtaining a spherical curve c_{3D}, the green curve in Fig.2(c). For each red point $p \in V_s$ in Fig.2(c), its nearest point on c_{3D}, called its target points V_{ob}, can be found. These target points are in fact the objective normal vectors for the triangles whose normal images are corresponding to the red points.

IV. Normal-guided mesh deformation

In this section, we will introduce the details of how to deform a triangle mesh to make the difference between their new normal vectors and objective normal vectors as small as possible, meanwhile minimize its shape change, i.e., minimizing vertex position and face orientation change.

The deviation value between the objective normal vector and new face normal vector of a deformed triangle can be measured by:

$$\begin{aligned} minf(v_1',v_2',v_3') &= \omega(f_1+f_2)+f_3+f_4+f_5 \quad (2)\\ f_1 &= \left|n_{ob}\cdot(v_2'-v_1')-0\right|^2 \\ f_2 &= \left|n_{ob}\cdot(v_3'-v_1')-0\right|^2 \\ f_3 &= \| (v_2'-v_1')-(v_2-v_1) \|_2^2 \\ f_4 &= \| (v_3'-v_1')-(v_3-v_1) \|_2^2 \\ f_5 &= \frac{1}{3}\| (v_2'-v_1')-(v_2-v_1) \|_2^2 \end{aligned}$$

Here, $v_1,v_2,v_3,v_1',v_2',v_3'$ are triangle vertices before and after deformation, respectively; $\| v \|_2^2$ is vector 2-norm; n_{ob} is the triangle's objective unit normal vector; ω is a positive weight value, which allows user to balance the developability and shape change. Intuitively, the larger the ω value is, the more

developable mesh surface we can obtain, and of course the more shape changes.

For effectiveness, rewrite Eq.(3) in matrix form, as

$$A_i = \begin{bmatrix} -\omega \cdot n_{ob} & \omega \cdot n_{ob} & 0 \\ -\omega \cdot n_{ob} & 0 & \omega \cdot n_{ob} \\ -I_{3\times3} & I_{3\times3} & Z_{3\times3} \\ -I_{3\times3} & Z_{3\times3} & I_{3\times3} \\ \frac{1}{3}I_{3\times3} & \frac{1}{3}I_{3\times3} & \frac{1}{3}I_{3\times3} \end{bmatrix}, \quad x_i = \begin{bmatrix} v_1' \\ v_2' \\ v_3' \end{bmatrix},$$

$$c_i = \begin{bmatrix} 0 \\ 0 \\ v_2 - v_1 \\ v_3 - v_1 \\ \frac{1}{3}(v_1 + v_2 + v_3) \end{bmatrix} \tag{3}$$

$Z_{3\times3}$, $I_{3\times3}$ denote 3×3 zero and identity matrix, respectively; A_i called triangle's coefficient matrix which only depends on triangle's objective normal and weights information; c_i only depends on triangle vertex position before deformation. The properties of A_i and c_i are critical for us to convert the nonlinear system to a linear system exactly.

Additionally, the deformation described by Eq.(3) is a triangle-based deformation which cannot preserve mesh connectedness [10]. However, in a linear system $Ax = b$, if a vertex appears in column vector x more than one times, they can be merged. Thus, a triangle-based deformation can be converted into a vertex-based deformation which preserves the mesh connectedness.

Therefore, considering this conversion, the total objective function for a whole triangle mesh patch can be written as Eq.(4), the dimension of coefficient matrix, unknowns, and knowns vector are $11N_t \times 3N_v$, $3N_v \times 1$, $11N_t \times 1$ respectivly (N_v, N_t are vertex and triangle total number, respectively)

$$argmin_{v_1', v_2', \ldots, v_n'} \parallel Ax - c \parallel_2^2 \tag{4}$$

$$A = \begin{bmatrix} A_1 & Z & Z \\ Z & \ddots & Z \\ Z & Z & A_n \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Designer hopes a design tool can add some position constraints and control the shape change in a given tolerance, complying with original design intents. We adopt the idea of moving least squares (MLS) [15] since it not only can fix some vertexes, but also can alleviate vertex flip. The discrete weight function $\omega(v)$ is set as

$$\omega(v) = \frac{1}{min\_dist(v, constraints)^2 + \varepsilon^2} \tag{5}$$

$min\_dist(v, constraints)$ computes the minimal distance from an arbitrary vertex to a constraint-vertex set "constraints", and ε is a small value, 0.001. Eq. (4) can be rewritten as

$$argmin_{v_1', v_2', \ldots, v_n'} \parallel AWx - c \parallel_2^2 \tag{6}$$

W is a diagonal weight matrix whose diagonal elements are each vertex's weight value.

TABLE I
STATISTICAL DATA OF CURVED CONE IN FIG.3 ($\#(v) = 720$, $\#(f) = 1280$, $E_{before}^{mean} = 2.498e^{-3}$, $AE = 0.1169$)

E_{Allow}	ω	ΔE^{mean}	t_D	t_I	t_T	
$0.02 \cdot AE$	15	$2.48e^{-3}$	-0.57%	5.58	102.3	107.88
$10 \cdot AE$	5	$8.03e^{-4}$	-67.8%	5.57	44.18	49.75

TABLE II
STATISTICAL DATA OF SKIRT IN FIG.4 ($\#(v) = 986$, $\#(f) = 1848$, $E_{before}^{mean} = 4.91e^{-3}$, $AE = 0.1113$, $\omega = 5$)

E_{Allow}	E_{after}^{mean}	ΔE^{mean}	t_D	t_I	t_T
$0.33 \cdot AE$	$4.86e^{-3}$	-0.99%	9.88	14.18	24.06
$10 \cdot AE$	$8.03e^{-4}$	-67.8%	10.23	1.39	11.62

And the error control can be easily achieved by checking the vertex increment Δv_i in each iteration, if $|\Delta v_i| > \delta$, then $\Delta v_i = \delta$. δ is a given threshold.

Using the MLS to compute the minimal value Eq.(6), setting its gradient to zero, we get

$$WA^T AWx = WA^T c \tag{7}$$

Since $WA^T AW$ is a sparse, positive, definite, symmetric matrix, we can perform the Cholesky decomposition and back substitution to efficiently solve it.

V. EXPERIMENTS AND RESULTS

We implement a prototype system on PC (Intel core (TM) 2, 2.13GHz, RAM 2GB), which repeatedly updates all the mesh vertices until the iterative number, $MaxIter$, exceeds a certain value, or, the total mean curvature decrease ratio, ΔE^{mean}, between two consecutive loops is within a threshold,ΔC_{max}^{mean}, 0.001%. E_{before}^{mean}, E_{after}^{mean} are used to store the mean curvature of before and after deformed mesh in each iterative step.

In our implementation, the sparse matrix decomposition is decomposed only once at the first step since it is a constant matrix (see Eq.7). And in its following iterative steps, only vectors x and c are updated acoording to Eq.3 on deformed coordinates. Thus, the total time cost t_T consists of sparse matrix ($WA^T AW$) decomposition time t_D and iterative time t_I.

The iterative time cost mainly depends on the algorithm's convergency speed which is also affected by several factors. one critical factor is the error allowance, E_{Allow}, which means each edge can vary how much in their position and orientation to achieve their personal objectives. We think this may depend on the average edge length, AE, in each mesh surface path. Another factor is the weight value ω, mentioned in Eq.3. (*Besides, the following annotation symbols are used: #(v or f): number of vertex or face;*).

All the aspects mentioned above of our algorithms are tested by three experiments.

Fig.3 illustrates a curved cone which can be deformed into a completely developable surface. In this case, table I shows that when the E_{Allow} is larger, not only the algorithm convegernecy speed is much faster, but also the developability is improved in a greate quantity.

TABLE III
STATISTICAL DATA OF WETSUIT MODEL IN FIG.5 ($E_{Allow} = 0.33 \cdot AE$, $\omega = 10$)

patches	$\#(v)$	$\#(f)$	AE	E^{mean}_{before}	E^{mean}_{after}	ΔE^{mean}	t_D	t_I	t_T
1,8	167	270	1.311	$4.47e^{-3}$	$4.35e^{-3}$	-2.55%	3.11	3.89	7.0
2,9	596	1074	1.637	$4.948e^{-3}$	$4.941e^{-3}$	-0.17%	4.56	31.23	35.79
3,10	193	331	1.573	$4.79e^{-3}$	$4.77e^{-3}$	-0.42%	3.63	32.85	36.48
4,11	255	448	2.05	$9.31e^{-3}$	$9.24e^{-3}$	-0.75%	4.27	24.06	28.33
5,12	250	409	1.55	$7.88e^{-3}$	$7.75e^{-3}$	-1.63%	4.16	21.83	25.99
6,13	481	811	1.73	$3.81e^{-3}$	$3.78e^{-3}$	-0.68%	4.11	2.20	6.31
7,14	431	771	1.99	$6.62e^{-3}$	$6.51e^{-3}$	-1.71%	3.85	32.17	36.02

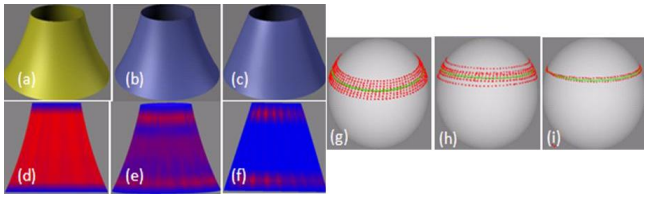

Fig. 3. Curved cone deform with error control and position constraints at the boundary, (a) original model; (b, c) deformed model with $E_{Allow} = 0.02 \cdot AE$, $E_{Allow} = 10 \cdot AE$ and their associated curva-ture distribution (d, e, f) and corresponding normal image (g, h, i).

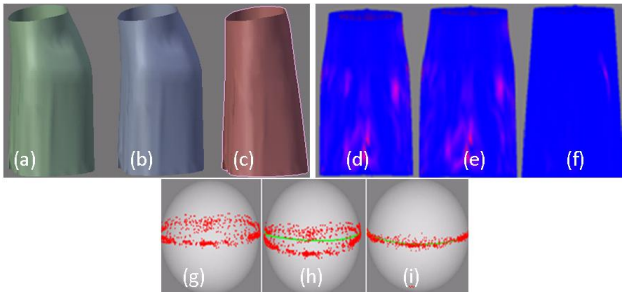

Fig. 4. Skirt model with boundary vertex constraints and error control. (a, b, c) Original model, deformed model of $E_{Allow} = 0.33 \cdot AE$, deformed model of $E_{Allow} = 10 \cdot AE$; (d, e, f) their associated curvature distribution; (g, h, i) their corresponding Gauss map image.

Fig.4 is a skirt surface with wrinkles which not suitable to fit by only one single patch. However, we use it test our algorithm's robustness. From Fig.4(b,c), we can conclude large E_{Allow} cannot preserve features in the original model. This is why we need this error control parameter E_{Allow}.

Fig.5 is a wetsuit which is segmented into pieces, then apply our algorithm to improve each patch's developability separately with vertex constraints at the boundary vertex and error control E_{Allow} all set to $1/300$ of AE. The efficient results are shown in Fig.5 and its statistical data listed in Table III.

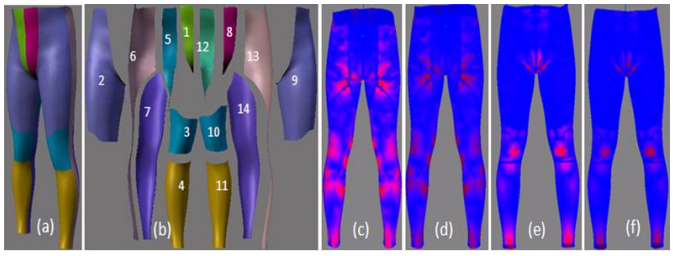

Fig. 5. Wetsuit model (a) Original model; (b) separated patches with patch number; curvature distribution of original model and deformed model, respectively in back view(c, d) and front view (e, f).

VI. CONCLUSION

We use a new developability measurement for mesh surface using its discrete normal image space. Based on this measurement, a new normal guided mesh deformation method with position constraints and error control is proposed. The originally highly non-linear problem is converted into a linear problem which can be solved efficiently. However, several aspects are still not deeply explored. Some other factors which influences the algorithm convergency speed, etc.

REFERENCES

[1] C. H. Chu, C. C. L. Wang, and C. R. Tsai. Computer aided geometric design of strip using developable bezier patches. *Computer in Industry*, 59:601–611, 2008.

[2] C.H. Chu and C.H. Séquin. Developable bézier patches: Properties and design. *Computer Aided Design*, 34(7):511–527, 2002.

[3] P. Decaudin, D. Julius, J. Wither, L. Boissieux, A. Sheffer, and M. Cani. Virtual garments: a fully geometric approach for clothing design. *Computer Graphics Forum*, 25(3), 2006.

[4] M. doCarmo. *Differential geometry of curves and surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[5] L. Fernàndez-Jambrina. B-spline control nets for developable surfaces. *Computer Aided Geometric Design*, 24(4):189–199, 2007.

[6] W. Frey. Boundary triangulations approximating developable surfaces that interpolate a closed space curve. *International Journal of Foundations of Computer science*, 13:285–302, 2002.

[7] A. A. Goshtasby. Fitting parametric curves to dense and noisy points. In *Int. Conf. Curves and Surfaces*, pages 227–237, 1999.

[8] D. Julius, V. Kraevoy, and A. Sheffer. D-charts: quasi-developable mesh segmentation. *Computer Graphics Forum*, 24(3):581–590, 2005.

[9] Y. Liu, H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang. Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics*, 25(3):681–689, 2006.

[10] W. R. Sumner and J. Popovic. Deformation transfer for triangle meshes. In *Inter. Conf. on Computer Graphics and Interactive Techniques*, pages 399–405, 2004.

[11] K. Tang and M. Chen. Quasi-developable mesh surface interpolation via mesh deformation. *IEEE Trans. on Visualization and Computer Graphics*, 15(3):518–528, 2009.

[12] C. C. L. Wang, S. S. F. Smith, and M. M. F. Yuen. Surface flattening based on energy model. *Computer Aided Design*, 34(11):823–833, 2002.

[13] C. C. L. Wang and K. Tang. Optimal boundary triangulations of an interpolating ruled surface. *ASME Journal of Computing and Information Science in Engineering*, 5(4):291–301, 2005.

[14] C. C. L. Wang, Y. Wang, and Y. F. M. Yuen. On increasing the developability of a trimmed nurbs surface. *Engineering with Computers*, 20:54–64, 2005.

[15] Y. Yoshioka, H. Masuda, and Y. Furukawa. A constrained least-squares approach to interactive mesh deformation. In *Conf. on shape modeling and applications*, pages 153–162, 2004.