

A Surface Development Method with Application in Footwear CAD/CAM

Angela Tam¹, A. Joneja², Kai Tang³ and Z. Yao⁴

^{1,2}Dept of IELM, HK University of Science & Technology, joneja@ust.hk

³Dept of ME, HKUST, mektang@ust.hk

⁴Dept of ACAE, Chinese University of HK, zyyao@mae.cuhk.edu.hk

ABSTRACT

Surface development is the process of mapping a given 3D surface into a 2D shape. The usual objective is to achieve this mapping while preserving isometry, or, for non-developable surfaces, to minimize the distortion. This problem has many applications in mechanical design, for example in ship-design, sheet-metal processing, textile design, etc. and is therefore well studied. However, previous approaches could not easily be adopted for our specific application, namely footwear design. Here we propose a new method, based on a simplified solid mechanics model. Preliminary experiments indicate that this approach improves upon some recently studied techniques. The approach also has an advantage that it can be extended quite naturally to handle composite surface patches where each patch has different material properties. The method has been implemented and some experimental results will be presented in the paper.

Keywords: surface development, solid mechanics, footwear design

1. INTRODUCTION

Surface development is the process that transforms a curved surface into a planar surface. A surface is defined as a developable surface if it can be flattened into planar surface without any distortion. The sufficient condition to test if a surface is developable is if its Gaussian curvature equals zero at all points. Contrarily, if the surface has non-zero Gaussian curvature at some point(s), then it must undergo some distortion when it is flattened. A surface is said to be *distorted* if the length of any continuous curve segment on it changes when it is developed. A non-developable surface cannot be flattened to a planar surface without some distortion (Figure 1a), or possibly introducing cuts (Figure 1b).

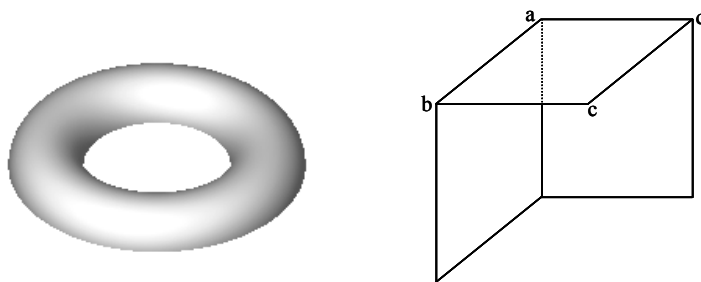


Fig. 1: (a) Toroidal surface (b) A surface that may be flattened by introducing a cut from a to any point on b-c-d.

Surface development has many engineering applications. Parts made by bending sheet metal must be developed before cutting them from sheets using punching or blanking. Panels used to construct boats etc. are also designed in this way. In the textile industry, development allows 3D clothes to be flattened into the shape of 2D patterns that are cut out of cloth rolls. Almost all methods for textile development start with tessellated approximation of the curved surface. The method to develop the non-developable surface is to cut the corresponding surface into pieces of near-developable surfaces, and then develop those developable patches (called *patterns*). One is often allowed to introduce cuts on the pattern; when the actual dress is made, two edges of the cut may be stitched together (such cuts are called *darts*), or an additional piece of cloth may be introduced along the cuts (such cuts are called *gussets*). The amount of distortion is kept below a very low threshold value at all points on the surface. The application we are interested in is

Footwear CAD/CAM. Shoe uppers are made by stitching together leather (or synthetic) patterns. Traditionally, only a few pattern types were known that would allow development without too much distortion. In other words, patterns of these shapes would allow stitching along acceptable lines on the shoe upper, and provide an upper with an acceptable level of wrinkling. Therefore classical shoe upper design has only a few standard pattern shapes. With an increase in demand for design variations, this restriction must be relaxed, but still provide aesthetically pleasing and functionally useful shoes.

This paper introduces a new algorithm for surface development useful in generating 2D patterns from shape profiles designed on a 3D surface of a shoe last by a shoe designer.

2. BACKGROUND

Due to its importance in various engineering applications, several approaches have been used to solve this problem. We summarize these methods here, before commenting on why yet another technique is proposed in this paper.

2.1 Segmentation Based Methods

One of the methods to develop non-developable surfaces is to decompose the curved surface into simpler surfaces and then approximate those simpler surfaces with developable surfaces. Finally, the approximated developable surfaces are unwrapped or unrolled into planar surfaces. Both G. Elber [3] and J. Hoschek [7], used this approach. G. Elber [3], proposed to use developable ruled surfaces to approximate the curved surface. A tolerance is set and the curved surface is approximated by developable ruled surfaces within this tolerance value. Unrolling the ruled surfaces onto a plane then generated the required patterns. J. Hoschek [7] suggested using partial conical and cylindrical surfaces to approximate surfaces of revolution. He also extended the method to approximate a given translation surface (a surface that is created by sweeping a 2D curve along a curved path), by a set of developable surfaces.

2.2 Energy Based Models

Many of the approaches for development use some variation of energy-based models. Usually, the first step of energy-based methods is to create a tessellation of the surface, the most convenient scheme being triangulation. The triangulated 3D surface is then mapped onto a plane. The mapping method is different for different energy-based models. The difference of the intrinsic energy in the triangulated surface and the mapped surface is calculated based on the proposed energy model. Furthermore, a threshold value is set to denote acceptable level of distortion energy in the developed pattern. The development process can then be modeled as an energy minimization problem; if the algorithm is iterative, then any solution with energy lower than the threshold is an acceptable pattern, terminating the algorithm.

J. McCartney et al. [9], proposed an algorithm to perform the development by considering the strain energy in the edges in a triangulated surface. The development process begins with a Delaunay triangulation of the surface. Therefore, the triangulation is not symmetrical but the vertices of triangle were well distributed. An energy-based model is then implemented as the core of the development process by considering each edge of triangles as a linear link; the entire network of links comprised the surface. The energy of the deformed surface is approximated by the sum of the extension/compression energy of all links forming the surface (i.e. shear is ignored). An ideal development would map the 3D shape to a 2D shape that could preserve the length of each link. Such preservation is not possible for non-developable surfaces, so energy relaxation is needed. McCartney proposed a simple iterative procedure for energy relaxation: for each node, try to move along each of the four orthogonal directions incrementally by some fixed small distance, and select the movement that can result a maximum energy reduction; if no movement reduces the energy, then try the next node. Every node movement is tested in order to search for a maximum energy reduction in each step of the iteration. If no energy reduction is achieved at a stage, with a large residual energy in the mapped surface, then darts are introduced to continue the relaxation.

2.3 Physics Based Models

Fan and Yuen [4] studied a variation of the energy-based models called physics based model for development. The same group later refined this formulation [10]. A uniform triangulation of the curved surface is used in their approach. The tessellation was mapped onto a plane to yield an initial planar shape with a topologically identical triangulation. Each edge, or link, was treated as a spring, and each vertex had a mass assigned to it based on the area of the neighboring triangles. Additional links, or springs, were added to approximate the shear energy. The initial planar surface was then modified iteratively according to the forces found by the spring-mass model. Darts are automatically inserted by the algorithm when the accuracy of the developed surface is low.

2.4 Geometry Based Models

Another category of techniques uses some differential properties of the surface. In this approach, during the flattening of a surface from 3D to 2D, some geometric properties are preserved. Examples of such properties include the geodesic curvature, arc lengths of some curves on the surface, or node angles on a network of curves on the surface. Such methods attempt to preserve local isometry on the surface. Faux and Pratt [5] note that the geodesic curvature can be used for surface development. J.R. Manning [8] proposed a method to flatten the surface based on a concept called the isometric tree. A series of geodesic curves called spines are drawn along the surface; several curves called branches are generated transversely between the spines. Development is achieved by first mapping the spines into lines on a plane. The relation between the spines is maintained by ensuring that the branches in the corresponding flattening are at the same angle as the corresponding tangential angle on the 3D surface. B.K. Hinds *et al.* [6] proposed a method to flatten the 3D surface by considering the Gaussian curvature of the surface. Their method is a generalization of Manning's, but their implementation only considered elliptic and hyperbolic-curvature surfaces. C. Bennis *et al.* [2] proposed a technique that considered the preservation of geodesic curvature along isoparametric curves. We start with an initial curve, and consider the geodesic curvature at each point in both u and v directions and the projected cross-angles between these two directions. The gradient of distortions for the developed surface is calculated. The distortion is then evenly distributed on the flattening region. The approach sometimes leads to gaps and overlaps along the isoparametric curves; also, the development is sensitive to the parameterization.

P.N. Azariadis *et al.* [1] proposed two methods for surface development also by preservation of the geodesic curvature of the surface curves. Consider that a curve is projected onto the tangent plane at the node points giving a planar curve. The preservation of the geodesic curvature of the surface curves implies the preservation of angle between two successive edges of the planar curve. The curved surface is triangulated. The nodes of the triangles are used as the vertices for the angle preservation constraint. They constructed an energy function that embodied the geodesic curvature at every point on the planar development. The overall geodesic curvature error of the planar development was formulated and the objective is to minimize the geodesic curvature error all over the planar development. The second method considers the local accuracy in the derived planar surface. By using this method, the strong dependency on parameterization of the curved surface was solved. To use this method, a curve on the surface has to be defined first; this curve can be one of the isoparametric curves or any user-defined curve on the surface. This curve would be the constraint added to the model, because the length and the geodesic curvature of the curve were preserved in the flattening process. The authors also constructed another corresponding energy function to minimize the overall geodesic curvature error of the planar development subjected to the constraints that the geodesic curvature of the selected curve have to be preserved and the length of the selected curve have to be preserved also. The formulation was a nonlinear minimization problem; the authors used a quadratic approximation of the Lagrangian and linearized the constraints to solve it. However, the time performance was poor. They considered using the fast conjugate gradient method to solve the formulation, but the accuracy of the result deteriorated. So, an appropriate tool to solve the formulation is still undiscovered.

We could not directly use any of these techniques for our application. In shoe design, location of cuts and gussets may be specified, and not necessarily derived from the last geometry. Secondly, materials may be anisotropic. Finally, the shoe upper may be a composite made up by stitching several patches of different materials. To handle these issues, we implemented an extension of McCartney's technique. However, instead of using relatively random attempts to perturb the shape of the triangles at each step, we were able to use a better engineering measure, namely surface strain energy, to guide the search. The initial implementation of this approach is described in this paper.

3. METHODOLOGY

It is quite common for shoe uppers to be made up of several patches, often of different materials. At the same time, it is conventional for shoe CAD/CAM systems to use an integrated view of the flattened upper in 2D for easier design specification. To handle this problem, we explored a solid mechanics based model for development. We briefly outline McCartney's method, since it captures the idea of several different methods [2, 4, 6, 9, 10]. The algorithm works on a triangulated tessellation of the 3D surface. A triangle in an interior location (approximately at the center of the patch) is used as a seed triangle. This triangle was then arbitrarily laid down on a 2D plane, so that it stores zero strain energy. Triangles that share an edge with this flattened triangle would be the next triangles to be flattened; these make up the active set of triangles. At any stage, the triangles making the surface belong to one of two types: at least one edge of a triangle in the active set has been flattened before; if the third node of the triangle has been flattened by previous flattening step, that triangle is constrained, and must be flattened by a constrained triangle flattening process. Otherwise, the flattening of the triangle is unconstrained. Once all triangles have been flattened, the resulting 2D shape

has a relatively large strain energy, which is reduced by a relaxation process. McCartney's relaxation process is: take each node in the flattened pattern and try to move it by a fixed, predetermined incremental distance Δd in one of four fixed orthogonal directions as shown in Fig. 2.

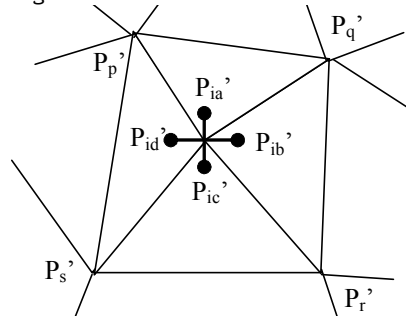


Fig. 2: Quantized perturbation of a node: {Pia, Pib, Pic, Pid}.

The corresponding energy reduction is calculated assuming that each edge of the triangle is a spring, connected to other edges at the vertex via revolute joints. The reduction in energy for each quantized move of each node is computed and stored. The move that yields the maximum energy reduction is selected and executed. The potential change in energy reduction due to moves of its neighboring nodes is then updated, and the greedy procedure is repeated until it converges (or for a fixed number of iterations).

There are three weaknesses in this method: (a) the step size Δd for each move is predetermined; (b) the direction of the relaxation step is arbitrarily constrained to remain axis-parallel; (c) the strain energy is underestimated since the shear components are ignored. In our flattening algorithm, the strain energy is computed by using a simple solid mechanics model considering elastic deformation energy of thin sheet of the shape of the triangle. The energy released during the iteration is by nodal displacement, but the principal strain directions determine the direction of displacement. Furthermore, the moving distance of the node is a function of the strain of the triangle and is constrained only by the geometry of the neighboring triangles to ensure that the topology of the graph induced by the triangulation does not change. Thus step-sizes are not quantized.

3.1 The Strain Energy Model

The strain energy is computed by first determining the principal directions. This allows a simple advantage: the principal directions can be used as a hint for determination of the direction of perturbation of a node during energy relaxation. Secondly, it is relatively straightforward to use standard sheet-strain formulas to compute the strain energy once the principal strains are known, since shear component vanishes along these directions. The knowledge of the principal strain is also useful in our algorithm for orienting the developed pattern on an anisotropic sheet. Consider a triangle ABC that has been transformed by a plane deformation as shown in Fig. 3 (a) & (b). The coordinates of each vertex are known, from which it is easy to compute the edge lengths (AB, BC, CA, A'B', B'C', C'A') and the interior angles ($\alpha_A, \alpha_B, \alpha_C, \alpha_A', \alpha_B', \alpha_C'$).

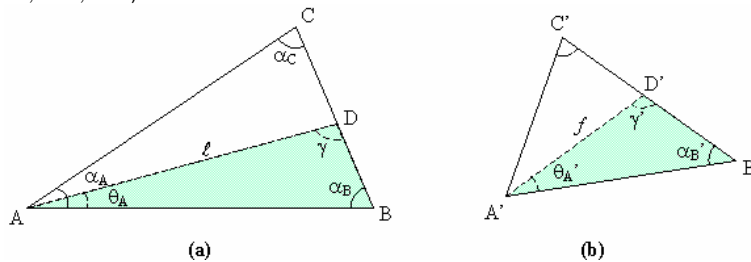


Fig. 3: Computing the strain energy: (a) Original Triangle (b) Transformed Triangle.

The length of AD calculated by sine rule in terms of θ_A : $l(\theta_A) = \frac{AB}{\sin(\theta_A + \alpha_B)} \times \sin \alpha_B$.

The length of A'D' calculated by sine rule and cosine rule in terms of θ_A :

$$f(\theta_A) = \sqrt{A'B'^2 + \left[\frac{B'C' \times AB \times \sin \theta_A}{BC \sin(\theta_A + \alpha_B)} \right]^2 - 2(A'B') \left[\frac{B'C' \times AB \times \sin \theta_A}{BC \sin(\theta_A + \alpha_B)} \right] \cos(\alpha_B')}$$

We know the length of AD and the length of A'D', now considering the strain of stripe at $\theta_A = \frac{\ell - f}{\ell} =$

$$1 - \frac{\sqrt{A'B'^2 + \left[\frac{B'C' \times AB \times \sin \theta_A}{BC \sin(\theta_A + \alpha_B)} \right]^2 - 2(A'B') \left[\frac{B'C' \times AB \times \sin \theta_A}{BC \sin(\theta_A + \alpha_B)} \right] \cos(\alpha_B')}}{\frac{AB \sin \alpha_B}{\sin(\theta_A + \alpha_B)}} \\ = 1 - \frac{\sqrt{A'B'^2 \sin^2(\theta_A + \alpha_B) + \left[\frac{B'C' \times AB \times \sin \theta_A}{BC} \right]^2 - \frac{2 \times (A'B' \times B'C' \times AB) \times \sin \theta_A \times \cos \alpha_B' \times \sin(\theta_A + \alpha_B)}{BC}}{AB \sin \alpha_B}$$

The principle strain direction is found by:

$$\frac{d(\text{strain})}{d\theta_A} = \frac{1}{AB \sin \alpha_B} \frac{d \left(\sqrt{A'B'^2 \sin^2(\theta_A + \alpha_B) + \left[\frac{B'C' \times AB}{BC} \right]^2 \sin^2 \theta_A - \left[\frac{2 \times A'B' \times B'C' \times AB \times \cos \alpha_B'}{BC} \right] \sin \theta_A \sin(\theta_A + \alpha_B)} \right)}{d(\theta_A)} = 0 \\ \Rightarrow -\frac{1}{AB \sin \alpha_B} \left[\frac{A'B'^2 \sin(2\theta_A + 2\alpha_B) + \left[\frac{B'C' \times AB}{BC} \right]^2 \sin(2\theta_A) - \left[\frac{2 \times A'B' \times B'C' \times AB \times \cos \alpha_B'}{BC} \right] \sin(2\theta_A + \alpha_B)}{2 \sqrt{A'B'^2 \sin^2(\theta_A + \alpha_B) + \left[\frac{B'C' \times AB}{BC} \right]^2 \sin^2 \theta_A - \left[\frac{2 \times A'B' \times B'C' \times AB \times \cos \alpha_B'}{BC} \right] \sin \theta_A \sin(\theta_A + \alpha_B)}} \right] = 0$$

$$\therefore 2\theta_A = \left[\tan^{-1} \left(\frac{2 \times A'B' \times B'C' \times AB \times BC \times \cos \alpha_B' \times \sin \alpha_B - A'B'^2 BC^2 \sin 2\alpha_B}{A'B'^2 BC^2 \cos 2\alpha_B + B'C'^2 AB^2 - 2 \times A'B' \times B'C' \times AB \times BC \times \cos \alpha_B' \times \cos \alpha_B} \right) \right]$$

$$\Rightarrow \theta_A = \left[\tan^{-1} \left(\frac{2 \times A'B' \times B'C' \times AB \times BC \times \cos \alpha_B' \times \sin \alpha_B - A'B'^2 BC^2 \sin 2\alpha_B}{A'B'^2 BC^2 \cos 2\alpha_B + B'C'^2 AB^2 - 2 \times A'B' \times B'C' \times AB \times BC \times \cos \alpha_B' \times \cos \alpha_B} \right) \right] / 2$$

For principle strain direction measured at vertex B, the equation becomes:

$$\theta_B = \left[\tan^{-1} \left(\frac{2 \times B'C' \times C'A' \times BC \times CA \times \cos \alpha_C' \times \sin \alpha_C - B'C'^2 CA^2 \sin 2\alpha_C}{B'C'^2 CA^2 \cos 2\alpha_C + C'A'^2 BC^2 - 2 \times B'C' \times C'A' \times BC \times CA \times \cos \alpha_C' \times \cos \alpha_C} \right) \right] / 2$$

Of course, it is possible to compute the principal directions via other vertices, e.g at vertex C, we get:

$$\theta_C = \left[\tan^{-1} \left(\frac{2 \times C'A' \times A'B' \times CA \times AB \times \cos \alpha_A' \times \sin \alpha_A - C'A'^2 AB^2 \sin 2\alpha_A}{C'A'^2 AB^2 \cos 2\alpha_A + A'B'^2 CA^2 - 2 \times C'A' \times A'B' \times CA \times AB \times \cos \alpha_A' \times \cos \alpha_A} \right) \right] / 2$$

The above formulation helps to determine the principal directions as the pair of orthogonal directions that correspond to the extreme values of the strain. Using vertex A, the corresponding principle strain direction, θ_A' , for the transformed triangle is given by: $\theta_A' = \frac{B'D'}{BD} \times \theta_A$, and the direction perpendicular to it. Once the principal directions are known, it is easy to compute the corresponding principal strain values. We then compute the strain energy U_i of the transformed

triangle i as: $U_i = \frac{1}{2} E(\varepsilon_1^2 + \varepsilon_2^2) t A_i$, where t is the (uniform, constant) thickness of the surface, and A_i is the area of the original triangle, and $\varepsilon_1, \varepsilon_2$ denote the strain along principal directions.

The total strain energy of the flattened surface is given by $U = \sum_{i=1}^n U_i$, where n is the number of triangles in the

approximation. Similarly, we have also derived the strain energy for anisotropic materials where the material exhibits different Young's modulus along the two orthogonal directions. This is common for some textiles/leathers with directional fibers. However the fiber orientations vary for each triangle (a) in the initial 3D→2D map, and (b) after each iteration. Hence we needed a separate method to handle anisotropic materials; due to space limitation, that model is not covered in this paper.

3.2 The Iterative Algorithm

The flattening algorithm has three steps.

Stage 1: Initial mapping:

The first step tessellates the given surface by triangulation. While several triangulating schemes will work equally well, in the implementation below we only use a simple format: a mesh of isoparametric curves is first generated, and each tiny 4-gon is triangulated by connecting one pair of diagonal corners. This scheme makes it easy to use a natural 2D map induced via mapping: $(u, v) \rightarrow (x, y)$. Further, the planar figure is scaled to equate its x -, y -diameters to the maximum length among the u -, v -curves respectively. This brings the (distorted) planar figure closer to the actual size, and speeds the subsequent iterative steps.

Stage 2: Coarse Movement of nodes

In this stage, an attempt is made to quickly relax the initial shape into a 'close-to-final' form by starting at the triangle located approximately in the middle of the mesh, and modifying its edges to their equivalent length in the 3D shape. This stage iterates once for each triangle, proceeding along a pattern as indicated by the numbering in Fig. 4. For each triangle except the first, any edge that has been modified in an earlier step is left unchanged -- therefore only some vertices may be allowed to move (the triangles in Fig. 4 that are un-numbered have all vertices constrained due to earlier steps). Secondly, vertices are allowed to move only as far as the topology of the mesh is unchanged. At the end of this stage, some triangles are completely distortion-free, and the remaining have significant distortion; for instance, the white triangles in Fig. 4 may indicate relaxed triangle, while the magenta ones have some residual strain.

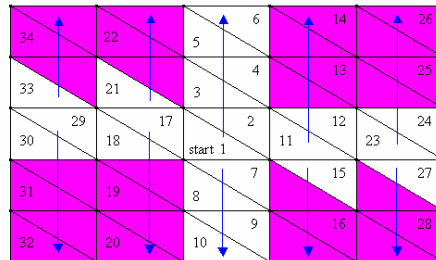


Fig. 4: Sequence of relaxing triangles in the coarse movement stage.

Stage 3: Fine Movement of nodes

This is the iterative, and usually the most time consuming stage of the algorithm. The energy relaxation algorithm moves one node at a given step, based on the direction and magnitude of the principal strains in the mapped triangle. Fig. 5 shows which node will be selected at each step, and the direction of the relaxation step. Let PQR be the original triangle and pqr be the mapped one. Assume (for illustration) that the two principal strains, ε_1 and ε_2 , are both compressive, and along the directions shown by the cross. Further, $\varepsilon_1 > \varepsilon_2$, hence the initial direction of the relaxation is selected along the direction of ε_1 . We select the vertex such that a vector placed at this vertex along the selected direction will pass through the triangle (vertex q in the figure).

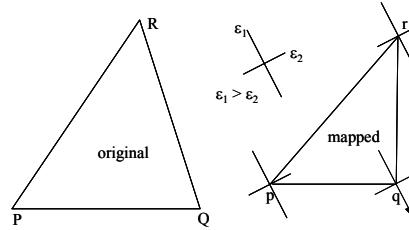


Fig. 5: Vertex selection for energy relaxation in a given triangle.

It is possible to show that this choice of relaxation provides (locally) the steepest gradient for energy relaxation. This is intuitively evident by the definition of the principal strains. We can show this analytically as follows. Let A , B and C be the sides of the original triangle, and a , b , c the corresponding sides of the mapped one. Assume further that the principal direction for the larger strain passes through the vertex at the top. We write these coordinates as (x, y) , in the coordinate frame shown below.

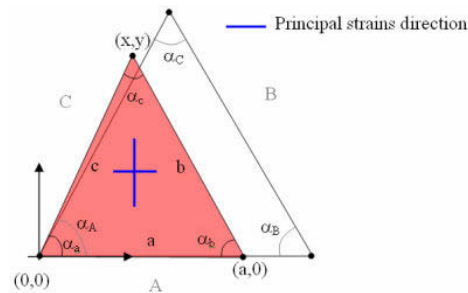


Fig. 6: Definition of the original triangle and the mapped triangle.

Thus, $a=a$; $b = \sqrt{(x-a)^2 + y^2}$, $c = \sqrt{x^2 + y^2}$; likewise, the angles α_a , α_b , and α_c are easily computed using the cosine rule. As discussed above, the candidate vertex is identified based on the strain; here we assume, without loss of generality, that it is the vertex (x, y) . So, strain1 along principal strain direction 1 is:

$$\text{Strain1} = 1 - \frac{\sqrt{a^2 \sin^2(\theta_1 + \alpha_B) + \left(\frac{bA \sin(\theta_1)}{B}\right)^2 + \frac{2abA \sin(\theta_1) \cos(\alpha_b) \sin(\theta_1 + \alpha_B)}{B}}{A \sin(\alpha_B)}$$

The principal strain direction θ_1 ,

$$\theta_1 = 0.5 \tan^{-1} \left(\frac{AB(2a^2 - 2xa) \sin(\alpha_B) + a^2 B^2 \sin(2\alpha_B)}{a^2 B^2 \cos(2\alpha_B) + (x^2 - 2xa + a^2 + y^2) A^2 - AB(2a^2 - 2xa) \cos(\alpha_B)} \right)$$

And strain2 along principle strain direction is:

$$\text{Strain2} = 1 - \frac{\sqrt{c^2 \sin^2(\theta_2 + \alpha_A) + \left(\frac{aC \sin(\theta_2)}{A}\right)^2 + \frac{2caC \sin(\theta_2) \cos(\alpha_a) \sin(\theta_2 + \alpha_A)}{B}}{C \sin(\alpha_A)}$$

Energy, $U \propto (\text{Strain}_1^2 + \text{Strain}_2^2)$

Clearly a numerical approach would be needed to find the globally optimal move for the vertex. Moreover, the motion of the vertex is constrained, since it is bounded by the edges of the other triangles neighboring it in the tessellation. However some cases were numerically verified. Consider a triangle with sides $A=10$, $B=12.083$, $C=12.083$, and a mapped triangle with $a=8$, and $(x, y) = (4, 11)$ as shown below. In this case, it is fairly clear that the principal strain is maximal along the positive Y-axis, and it is further easily verified that as the top vertex is moved along this direction, the energy first decreases until the original and mapped triangles are equal in height (principal strain becomes zero in this direction), and then increases again. The strain energy, U is given by:

$$\text{Energy, } U = \left(1 - \frac{\sqrt{a^2 \sin^2(\alpha_B)}}{A \sin(\alpha_B)}\right)^2 + \left(1 - \frac{\sqrt{\frac{x^2 + y^2 + a^2 C^2 \cos(\alpha_A)^2}{A^2} - \frac{2aC \cos(\alpha_A)x}{A}}}{C \sin(\alpha_A)}\right)^2$$

$$\frac{\partial \text{Energy}}{\partial x} = \frac{-\left(1 - \frac{\sqrt{\frac{x^2 + y^2 + a^2 C^2 \cos(\alpha_A)^2}{A^2} - \frac{2aC \cos(\alpha_A)x}{A}}}{C \sin(\alpha_A)}\right)}{\frac{\sqrt{\frac{x^2 + y^2 + a^2 C^2 \cos(\alpha_A)^2}{A^2} - \frac{2aC \cos(\alpha_A)x}{A}}}{C \sin(\alpha_A)}} * \left(2x - \frac{2aC \cos(\alpha_A)}{A}\right)$$

$$\frac{\partial \text{Energy}}{\partial y} = \frac{-2\left(1 - \frac{\sqrt{\frac{x^2 + y^2 + a^2 C^2 \cos(\alpha_A)^2}{A^2} - \frac{2aC \cos(\alpha_A)x}{A}}}{C \sin(\alpha_A)}\right)}{\frac{\sqrt{\frac{x^2 + y^2 + a^2 C^2 \cos(\alpha_A)^2}{A^2} - \frac{2aC \cos(\alpha_A)x}{A}}}{C \sin(\alpha_A)}} * y$$

It is verified numerically that at this point the partial derivatives of Energy, $\partial \text{Energy} / \partial x$ and $\partial \text{Energy} / \partial y$ vanish, indicating that it is a local extreme point. Fig. 7(b) shows a MATLAB plot of change of energy for deviations of x, y . The direction of steepest descent on the plot agrees with the direction of the perturbation suggested by the principal direction.

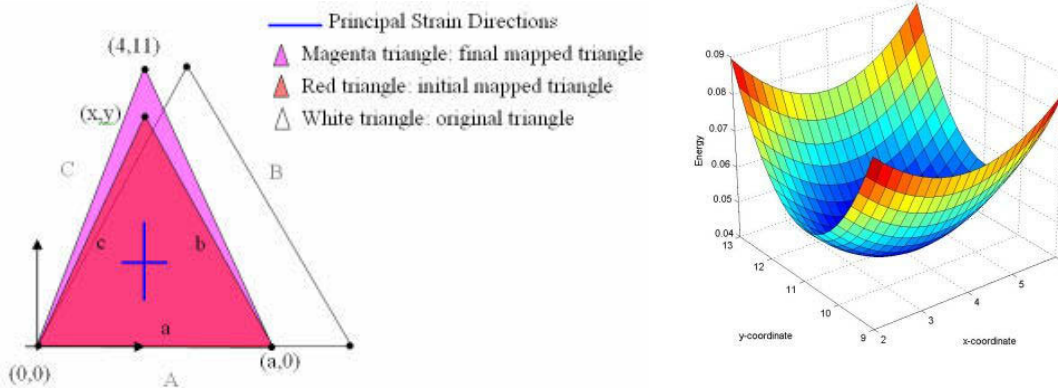


Fig. 7: (a) Example showing numerical verification of the choice of vertex move (b) Energy plot.

The figure below illustrates how the vertices are moved during energy relaxation. Vertex v_i is being considered for a move, which can be along one of the two principal strain directions (indicated in blue arrows). The coordinates of the moved vertex, v_i' , are calculated as:

$$(x_i \pm (\ell - f) \times \text{strain}_1 \times \cos(\text{abs_pdir}_1), y_i \pm (\ell - f) \times \text{strain}_1 \times \sin(\text{abs_pdir}_1)),$$

or

$$(x_i \pm (\ell - f) \times \text{strain}_2 \times \cos(\text{abs_pdir}_2), y_i \pm (\ell - f) \times \text{strain}_2 \times \sin(\text{abs_pdir}_2))$$

where v_i is at (x_i, y_i) , strain_1 and strain_2 are strains along principal strain directions 1 and 2 respectively.

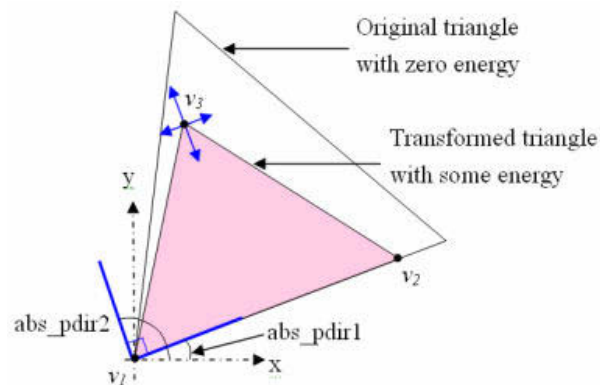


Fig. 8: Illustration of energy relaxation processes.

The vertex is moved iteratively, updating the principal direction after each move, and re-assessing the step size to be attempted; iterations stop when no reduction in energy is achievable.

Several simple methods of iterating through the triangles were implemented, including selecting row-by-row moving in a zig-zag pattern, or by using a greedy approach, namely selecting the currently most stressed triangle next. We have not found any basis for preference among these, and therefore the choice is left to the system user.

4. IMPLEMENTATION AND COMPARISON

The approach was implemented, and tested for various designed surfaces. We also implemented a version of McCartney's algorithm for some benchmarking. Here we show the results for two simple examples. The first example is a front pattern of a ladies mule. The second is one half of a men's dress shoe. In the results below, the colors on the triangles indicate different types (compression or extension) and levels of residual strain energy.

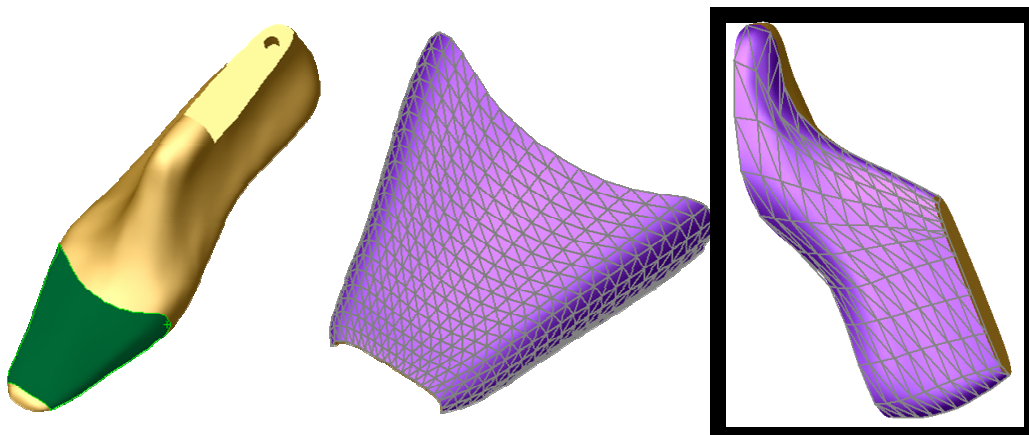
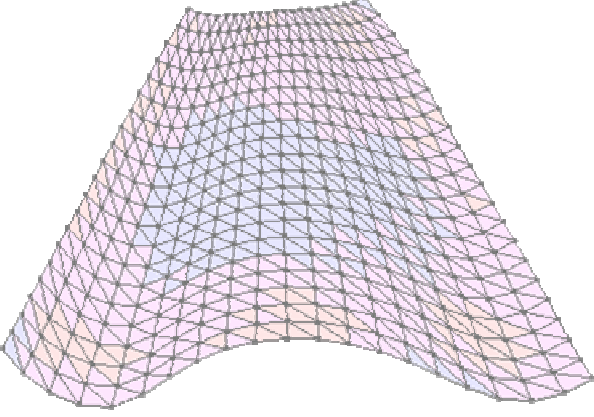
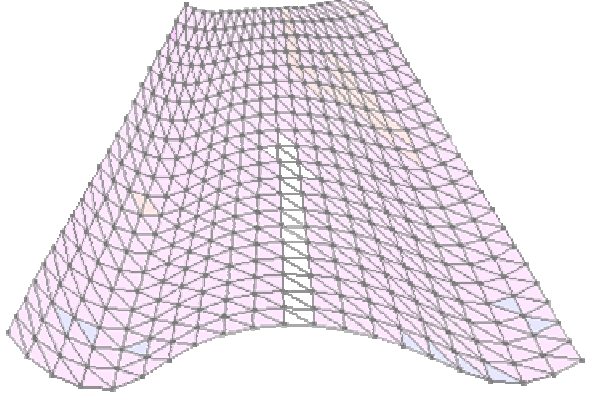
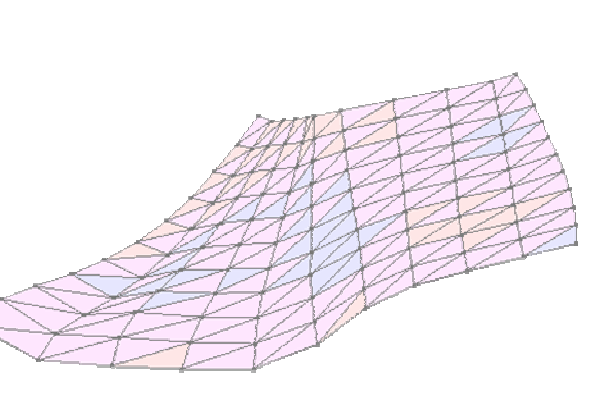
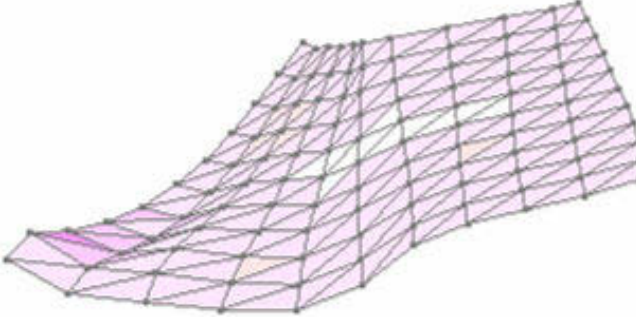


Fig. 9: (a) Front pattern on a ladies mule (b) Triangulation of its 3D surface (c) Second example.

	<p>Our approach:</p> <p>Computational time = 420s No. of iterations = 309 Final total area = 5603.12mm² Difference in area = 7.30mm² Initial energy = 2844.6809Nmm Utotal = 0.3993Nmm Total energy reduced 99.9860%</p>
	<p>McCartneys method</p> <p>Incremental nodal moving = 0.001mm Final Strain Energy = 2.4971Nmm Area of the original surface = 5610.42mm² Area of the flattened pattern = 5583.07mm² Difference in area = 27.35mm² Area reduced 0.4874% Total energy reduced 99.9122% Computational time = 32.367sec</p>
	<p>Our approach:</p> <p>Computational time = 248s No. of iterations = 286 Final total area = 18162mm² Difference in area = 47.0982mm² Initial energy = 19280.2Nmm Utotal = 8.52916Nmm Area reduced 0.258651% Total energy reduced 99.9558%</p>

	<p>McCartneys method</p> <p>Incremental nodal moving = 0.001mm</p> <p>Final Strain Energy = 80.2961Nmm</p> <p>Area of the original surface = 18209.1mm²</p> <p>Area of the flattened pattern = 17698.2mm²</p> <p>Difference in area = 510.96mm²</p> <p>Area reduced 2.8061%</p> <p>Total energy reduced 99.5835%</p> <p>Computational time = 39.337sec</p>
---	---

5. CONCLUDING REMARKS

From the results of several experiments, it is apparent that our new approach converges to a somewhat better result than McCartney's approach, although it takes a rather longer time to do so. The two main contributors to the longer convergence time are (a) higher number of iterations and (b) longer computation time per iteration. We are interested to investigate if we can apply a parallel processing algorithm (as opposed to one triangle per iteration) to improve the algorithm performance. Our approach has been tested for several different upper patterns on different types of shoes, and the program converges in approximately the same amount of time with similar levels of distortion. We are now looking into enhancing our approach to handle composite patches. This can be done fairly easily by assigning different strain modulus values to different triangles, depending on which patch on the surface it belongs to. As mentioned earlier, we have also implemented a version of our method that can handle anisotropic materials. These results will be discussed, possibly in an extended version of this paper in the near future.

6. ACKNOWLEDGEMENTS

Part of the work in this project was funded by CERG RGC grant No. HKUST 04E/614205.

7. REFERENCES

- [1] Azariadis, P. N.; Aspragathos, N.A.: Geodesic curvature preservation in surface flattening through constrained global optimization, *Computer-Aided Design*, 33, 2000, 581-591.
- [2] Bennis, C.; Vezien, J. M.; Iglesias, G.: Piecewise surface flattening for non-distorted texture mapping, *Computer Graphics*, 25(4), 1991, 237-246.
- [3] Elber, G.: Model fabrication using surface layout projection, *Computer-Aided Design*, 27(4), 283-291.
- [4] Fan, J.; Wang, Q.; Chen, S. F.; Yuen, M. F.; Chan, C. C.: A spring-mass model-based approach for warping cloth patterns on 3D objects, *Journal of Visualization and Computer Animation*, 9, 1998, 215-227.
- [5] Faux, I. D.; Pratt, M. J.: *Computational Geometry for Design and Manufacture*, England, Ellis Horwood Limited, 1979, 4, 116-120.
- [6] Hinds, B. K.; McCartney, J.; Woods, G.: Pattern development for 3D surfaces, *Computer-Aided Design*, 23(8), 583-592, 1991.
- [7] Hoschek, J.: Approximation of surfaces of revolution by developable surfaces, *Computer-Aided Design*, 30(10), 1998, 757-763.
- [8] Manning, J. R.: Computerized pattern cutting, *Computer-Aided Design*, 12(1), 43-47, 1980.
- [9] McCartney, J.; Hinds, B. K.; Seow, B. L.: The flattening of triangulated surfaces incorporating darts and gussets, *Computer-Aided Design*, 31(4), 249-260, 1999.
- [10] Wang, C. L.; Chen, S. F.; Fan, J.; Yuen, M. F.: Two-dimensional trimmed surface development using a physics-based model, *Proceedings of DETC'99, 25th ASME Design Automation Conference*, Paper No. DETC99/DAC-8634, Sept 1999.