Gap Filling Using a Blend of Morphing and Extension with Application in Footwear Design

Guo Li[1] and Ajay Joneja[2]

[1]Hong Kong University of Science & Technology, guoli@ust.hk
[2]Hong Kong University of Science & Technology, joneja@ust.hk

ABSTRACT

Several techniques for filling gaps or holes in a sculptured surface are known; perhaps the best studied include morphs and surface blending functions. Both of these techniques have several desirable properties, as some shortcomings. Here we develop a new operator that creates a hole-filling surface by interpolating the outputs of morphing and blending functions. By controlling the interpolating function, more control can be exhibited over the resulting shape than is provided by either of the standard operations individually. Further, we also demonstrate a method to produce a 'fair' surface (in the traditional, energy minimization sense) interpolating two gap-filling surfaces, created respectively by morphing and blending (or extension). The technique was motivated by problems encountered in the CAD of footwear.

Keywords: Blending, Linear Morphing, Gap Filling, Surface extension

1. INTRODUCTION

The work described in this paper is motivated by a problem we encountered in CAD of footwear. Design of a shoe begins with design of the shoe last, the mould around which the shoe is constructed. Here we shall differentiate between two 'parts' of a shoe last: the toe part and the rear, or back part. Typically the geometry of rear part is controlled by comfort requirements so it is determined by geometry of foot; this part has less freedom for design. It is common for many companies to maintain a library of standard back-part designs (a library is required since the geometry depends on the heel height, footwear style, e.g. sandal, pump, boot etc., and perhaps the location of the target customers). However, much freedom is allowed in the shape of the toe, which is largely dictated by aesthetic concerns. A convenient design strategy is to first design the toe part, and then select a suitable rear part from library and 'combine' these two components to create the last. Another common technique used by practitioners is to use the toe shape of one style of shoe and adapt it to fit with the rear part of a different design to create new designs. Figure 1 shows a CAD model of a simple last, with the rear and the toe parts marked. In practice to give freedom for designer to smooth out whole shape, there is a gap between toe part and rear part, otherwise combining a toe and rear that come from different designs will result in a discontinuity between the two shapes at the interface. Thus, in figure 1, there would be a gap of approximately 10mm on each side of the plane shown separating the toe from the back. Here, we shall assume that the new toe has been scaled appropriately and located in the correct position with respect to the back part. This scaling and positioning may be done with some user interaction, since the new toe shape must satisfy comfort and aesthetic guidelines that are sometimes subjective. We must then create a surface in the gap between the toe and the back part, to get a complete surface describing the last. This paper concerns the development of a CAD operation that to perform this 'gap filling'.

There are several well studied techniques for filling gaps in surfaces: one is to do blending between fitted surfaces (usually toe surface and rear surface are fitted from 3D scanned data). A large amount of blending research in the past has concentrated on the creation of variable- or constant-radius fillet features in CAD models. These essentially result in exact or approximated (parametric) forms of canal surfaces or their restricted form, pipe surfaces [1-4]. A large class of blending models are based on some form or generalization of the notion of filleting, including the use of cyclides. Some researchers have also considered issues such as smoothness of the blending surface. We are aware of at least two different methods to do so. In [5], the smoothness of the blending surface was derived by filtering out all high frequency variations of the surface in the region of the blend. The tool used to do so was Fourier transforms. However, this work obviously requires the geometry of the underlying surface(s) to be known before the blend is applied. Another typical problem in generation of blends is the complexity of the geometry and topology in regions where

several components of the blending surface intersect. This occurs, for example, in the neighborhood of a vertex where the blending surfaces for all the incident edges of a part meet. To tackle this problem, a method to generate a minimum energy surface to form the blend has also been developed [6].

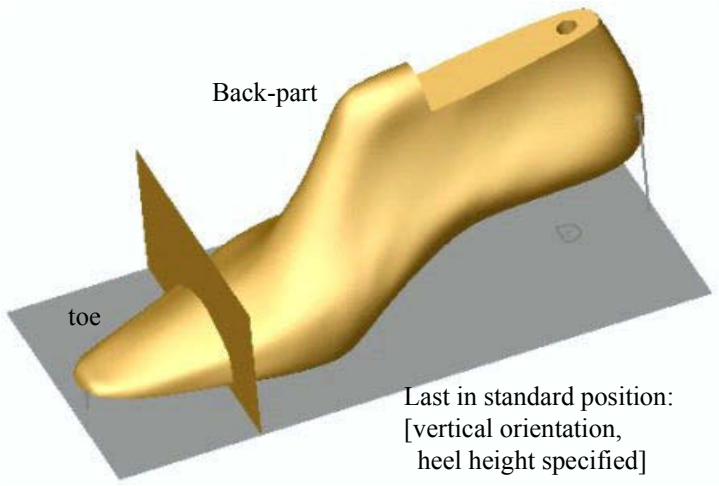

Fig. 1. A typical shoe last with the toe and rear parts marked

Another approach is that of generation of "fill" surfaces – namely a surface that smoothly interpolates a set of given curves forming a network. A simple case is when the interpolated curves form a closed loop that is marked as a boundary of the filling surface. At other times, the network is in the form of a series of curves, and a skinning operation can be applied [7]. However, this approach failed to give sufficient control on the resulting surface, partly due to the lack of appropriate guide curves bridging the gap. Smooth interpolating fill surfaces have also been generated using techniques that are based on subdivision surfaces [8]. However, this approach yielded surfaces for which subsequent shape modification of the shoe last was stunted due to lack of tools.

Another approach can be conceived based on morphing. While most morphing research has been done in the context of graphics, nevertheless the basic strategy is that of interpolation. In particular, interpolating functions (called warp functions) that are developed for vector representations of objects e.g., in [9] have possible relevance to our problem. In most cases, morphing has been applied to 2D images or shapes. Various methods to study 3D morphing have also been developed (for a good survey, please see [10]). In several of these approaches, the objects are treated as elastic bodies, and in each interpolating step, a distortion is imposed based on work minimization. For the basics of this idea, the reader is referred to [11].

Both, morphing (including linear and non-linear) and surface blending operations were applied to numerous examples in our problem domain. While either technique yields acceptable results in many cases, some deficiencies were found. The following description of the approaches sets up the motivation of our problem. The inputs to the gap filling operators are two (structured) sets of point clouds, one for the toe part, and one for the back, scaled and located in their expected final position with respect to each other in a global coordinate frame. For blending, the point clouds for the toe and the back parts were first fitted (within a specified tolerance) with B-spline surfaces. Then the gap between the boundaries of these two surfaces was joined by a blending surface. G^2 continuity was used when possible. In many cases, the quality of the resulting surface was undesirable: the connecting (intermediate) surface created to join the back part with the toe tends to "lose" the shape characteristics of the surfaces being connected. On the other hand, if a purely morphing based approach (using linear or non-linear warp functions) was applied, then the intermediate surface exhibits poor geometric continuity in the neighborhood of the boundary with the toe and the back part surfaces. Examples of these issues are shown in the figures below. Figure 2 shows the top view of an application of linear morphing. The toe and back-part are the point clouds in green color, while the morphed point clouds filling the gap are in magenta. A neighborhood of the boundary is enlarged to show the discontinuity problem. In figure 3, the intermediate surface is generated by the use of a G^1 blended surface. The resulting surface is smooth, but bulges out

too much on one side side, and squeezes in excessively on the other side. Further, providing additional control on the blended shape (beyond the continuity at boundaries) often results in constrained solutions that yield poor surface quality in other regions.

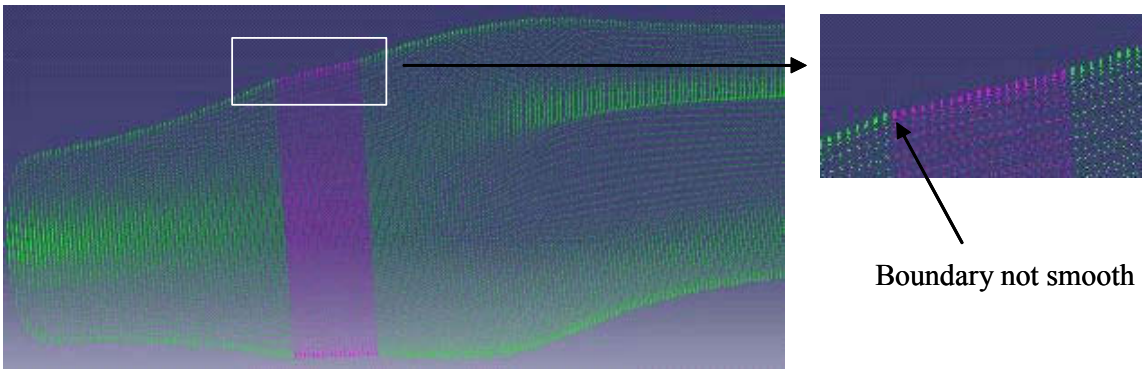

Fig. 2. Example of problem with morphing

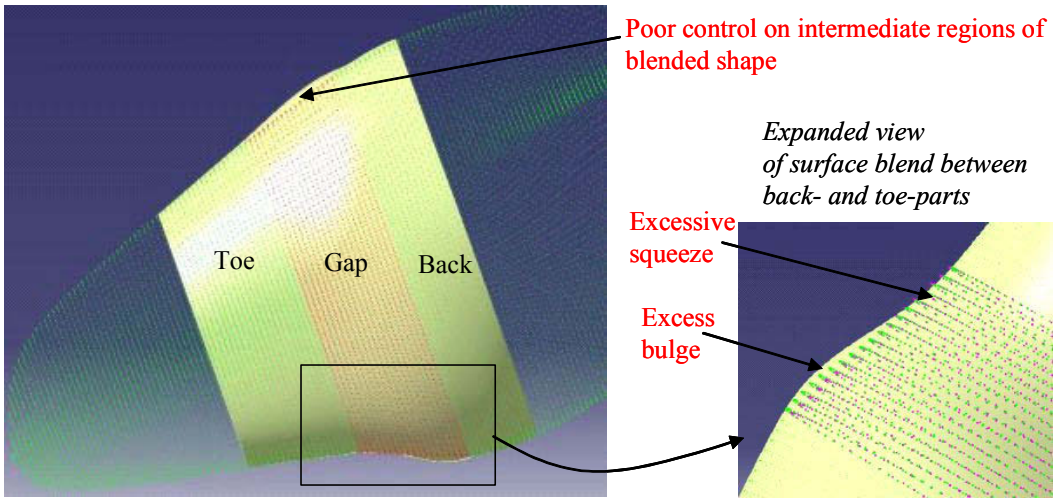

Fig. 3. Examples of problems (in red text) with blending

In order to provide more flexibility to the designer, we develop a new operator in this paper that allows arbitrary combinations of extended and morphed surfaces. Section 2 describes in detail how the new operator is constructed. The following section discusses a method to adapt a fairing-based approach to automatically compute the function blending the morphing and extension surfaces. Finally, the results of a preliminary implementation are given to show the use of this operator.

2. METHODOLOGY

Before setting up the details of the problem, we first provide some concrete context that is specific to the domain of shoe last design. This background gives a concrete context for our examples, though some of the initial detailed steps in relative positioning of the two surfaces are not necessarily relevant in the general context. Further, our model is somewhat dictated by the structure of the input data we shall assume; thus its application to the general gap filling will require some extensions that are non-trivial, at least from the implementation point of view.

The geometry of each last is stored as a structured point cloud. The back part and the toe geometries (from different shoe lasts) are generated by first placing each last model in a standard position such that the cutting plane

separating the two components is vertical. The initial models are assumed to be in the form of structured point clouds, such that the data is made up of points in layers; each layer is in a plane parallel to the cut making the gap. When two components that originally belonged to different lasts are selected for the merge operation, the toe part is scaled appropriately and located in the correct position with respect to the back part (maintaining a gap of approximately the same size). This scaling and positioning is done with some user interaction, since the new toe shape must satisfy comfort and aesthetic guidelines that are sometimes subjective. Thus, the data is structured in the sense that the underlying surfaces of the lasts are sampled in a series of layers, forming a sequence of z-constant contours. Each contour is made up an equal number of points. It is possible to convert other formats of inputs into this format (although this introduces some loss of accuracy due to sampling and subsequent fitting). A series of planar, interpolating contours are created in the gap; each of these contours will be defined as a series of points (see figure 4). As described earlier, the final surface is a blend between a morphed and an extension surface filling the gap.

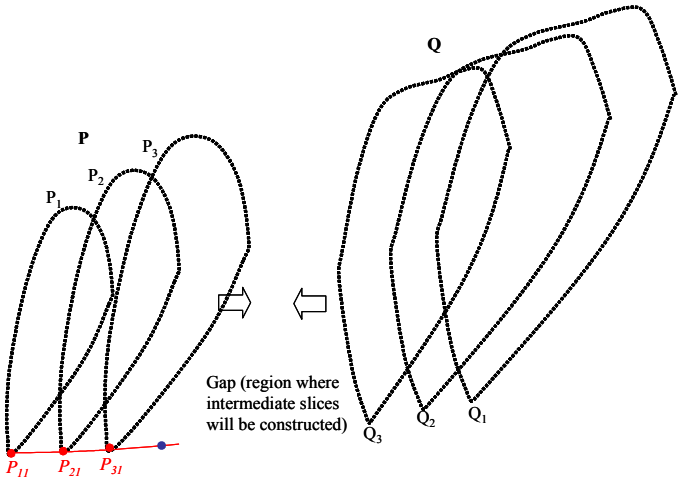

Fig. 4. Slices of toe and back parts with gap

2.1. The Morphed Surface

It was found that linear morphing was the most efficient method that preserved the shape characteristics in most cases. Morphing techniques are well known, and we shall only briefly sketch the details of generating the point cloud for any intermediate slice, Sj, in the gap. The morph is between the boundary slice of the toe (P3 in Figure 3) and that of the back (Q3). An extreme point is found on P3, say P31. Its corresponding point Q31 is found by a nearest neighbor search among the points of Q3. The corresponding point S31 is computed is just the intersection of the line segment P31-Q31 and the plane $z = Z_{Sj}$. Each subsequent point on Sj is computed by interpolating between each pair of subsequent points on P3 and Q3 respectively.

2.2. The Extension Surface

Two possible methods of generating extension (blended surface) in the gap were tried. In the first case, a surface ST was first found, interpolating the point cloud making up the toe part; likewise, an interpolating surface SB was found interpolating the point cloud of the back part. Then a blending surface between ST and SB is found, with G1 continuity at the boundaries. By taking appropriate planar sections and sampling, a suitable point-cloud can then be found for the blended surface. While this approach is relatively easy to implement using a commercial CAD package, we found that on several occasions, the quality of the intermediate surface was very poor, with excessive wrinkling and/or warping.
A discrete surface extension technique was therefore developed as an alternative. We discuss below how to generate one slice of data, say, S, past slice P3, using this discrete extension. Geometric continuity is well defined for parametric surfaces [12-16], but continuity conditions for discrete case have received less attention. Consider two surfaces, B and C, depicted in the parameter space as in Figure 5. The conditions for G1 and G2 continuity are given as:

G1: $$\alpha(v)\frac{\partial}{\partial u}B(u,v)\Big|_{u=0} + \beta(v)\frac{\partial}{\partial s}C(s,v)\Big|_{s=0} + \gamma(v)\frac{\partial}{\partial v}C(s,v)\Big|_{s=0} = 0 \quad (1)$$

G2: $$(\alpha(v))^2\frac{\partial^2}{\partial u^2}B(u,v)\Big|_{u=0} - (\beta(v))^2\frac{\partial^2}{\partial s^2}C(s,v)\Big|_{s=0} - 2\beta(v)\gamma(v)\frac{\partial^2}{\partial s\partial v}C(s,v)\Big|_{s=0} - (\gamma(v))^2\frac{\partial^2}{\partial v^2}C(u,v)\Big|_{u=0} = \delta(v)\frac{\partial}{\partial s}C(s,v)\Big|_{s=0} + \eta(v)\frac{\partial}{\partial v}C(u,v)\Big|_{u=0} \quad (2)$$

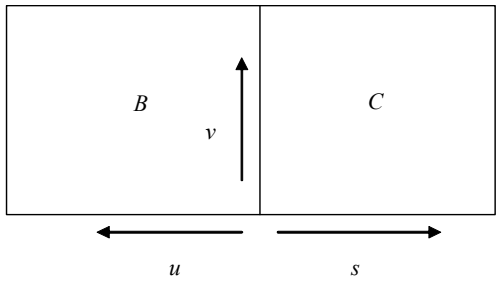

Fig. 5. Surfaces B and C in parameter space

Without losing generality, we can set $\alpha(v)$ as 1. We use a discrete, simplified form of the continuity conditions (1) and (2) to calculate extending slices. Note that while in each iteration, one extending slice, P_{ext} is computed; however, to compute it with G2 continuity we need also to construct a second extension slice to match the second derivatives. For convenience we denote original slices as P1, P2 and P3 as O, P and Q respectively. Assume that Zp-Zo = Zq-Zp = d; new slices, R and S are constructed on planes z=Zq+d/2 and z=Zq+d respectively.

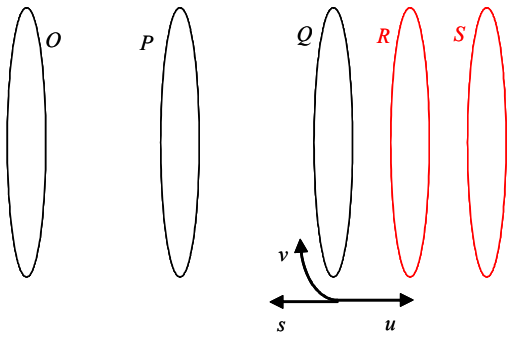

Fig. 6. Schematic of generation of slice S (and intermediate slice R)

Using finite difference we discretize G1 condition at point Q_i into:

$$\frac{(\overrightarrow{R_i} - \overrightarrow{Q_i})}{\Delta u} + \beta\frac{(\overrightarrow{P_i} - \overrightarrow{Q_i})}{2\Delta s} + \gamma\frac{(\overrightarrow{Q_{i+1}} - \overrightarrow{Q_{i-1}})}{2\Delta v} = 0 \quad (3)$$

For symmetry, let $\Delta u = \Delta s = k\Delta v$; from the z-coordinate we get $\beta = 1$. From (3) we can get $\overrightarrow{R_i}$ as:

$$\overrightarrow{R_i} = (\frac{3\overrightarrow{Q_i}}{2} - \frac{\overrightarrow{P_i}}{2}) - \gamma'(\overrightarrow{Q_{i+1}} - \overrightarrow{Q_{i-1}}) \quad (4)$$

in which $\gamma' = \gamma k / 2$ or $\gamma = 2\gamma' / k$. Since γ' can be chosen arbitrarily, there are infinite solutions for $\overrightarrow{R_i}$ (constrained to lie on a line). We fix it as the point closest to $\overrightarrow{Q_i}$:

$$\overrightarrow{R_i} - \overrightarrow{Q_i} = \frac{(\overrightarrow{Q_i} - \overrightarrow{P_i})}{2} - \gamma'(\overrightarrow{Q_{i+1}} - \overrightarrow{Q_{i-1}}) \cdot \left|\overrightarrow{R_i} - \overrightarrow{Q_i}\right| \text{ is minimized when}$$

$$\gamma' = \frac{(\overrightarrow{Q_i} - \overrightarrow{P_i}) \cdot (\overrightarrow{Q_{i+1}} - \overrightarrow{Q_{i-1}})}{2\left|\overrightarrow{Q_{i+1}} - \overrightarrow{Q_{i-1}}\right|^2} \tag{5}$$

Substituting (5) into (4), we compute $\overrightarrow{R_i}$ to get all points on the intermediate slice R.

The discrete form of (2) can be written as:

$$\frac{(\overrightarrow{S_i} + \overrightarrow{Q_i} - 2\overrightarrow{R_i})}{(\Delta u)^2} - \frac{(\overrightarrow{O_i} + \overrightarrow{Q_i} - 2\overrightarrow{P_i})}{4(\Delta s)^2} - \gamma \frac{(\overrightarrow{P_{i+1}} - \overrightarrow{P_{i-1}} - \overrightarrow{Q_{i+1}} + \overrightarrow{Q_{i-1}})}{4\Delta s \Delta v} - \gamma^2 \frac{(\overrightarrow{Q_{i+1}} + \overrightarrow{Q_{i-1}} - 2\overrightarrow{Q_i})}{(\Delta v)^2}$$
$$= \eta \frac{\overrightarrow{Q_{i+1}} - \overrightarrow{Q_{i-1}}}{2\Delta v} + \delta \frac{(\overrightarrow{P_i} - \overrightarrow{Q_i})}{2\Delta s} \tag{6}$$

By z coordinate, we have $\delta = 0$;

Substituting $\gamma = 2\gamma' / k$ into (6):

$$(\overrightarrow{S_i} + \overrightarrow{Q_i} - 2\overrightarrow{R_i}) - \frac{(\overrightarrow{O_i} + \overrightarrow{Q_i} - 2\overrightarrow{P_i})}{4} - \gamma' \frac{(\overrightarrow{P_{i+1}} - \overrightarrow{P_{i-1}} - \overrightarrow{Q_{i+1}} + \overrightarrow{Q_{i-1}})}{2} - 4\gamma'^2 (\overrightarrow{Q_{i+1}} + \overrightarrow{Q_{i-1}} - 2\overrightarrow{Q_i})$$
$$= \frac{\eta \cdot k \cdot \Delta u}{2} (\overrightarrow{Q_{i+1}} - \overrightarrow{Q_{i-1}})$$

Let $\eta' = \frac{\eta \cdot k \cdot \Delta u}{2}$ we simplify (6) into

$$(\overrightarrow{S_i} + \overrightarrow{Q_i} - 2\overrightarrow{R_i}) - \frac{(\overrightarrow{O_i} + \overrightarrow{Q_i} - 2\overrightarrow{P_i})}{4} - \gamma' \frac{(\overrightarrow{P_{i+1}} - \overrightarrow{P_{i-1}} - \overrightarrow{Q_{i+1}} + \overrightarrow{Q_{i-1}})}{2} - 4\gamma'^2 (\overrightarrow{Q_{i+1}} + \overrightarrow{Q_{i-1}} - 2\overrightarrow{Q_i})$$
$$= \eta'(Q_{i+1} - Q_{i-1})$$

η' can be randomly chosen so $\overrightarrow{S_i}$ is on a line (one freedom left);

As before, we minimize $\left|\overrightarrow{S_i} - \overrightarrow{R_i}\right|$ to fix η' ($\overrightarrow{S_i}$ is accordingly fixed); Let:

$$\vec{a} = (\overrightarrow{Q_i} - 2\overrightarrow{R_i}) - \frac{(\overrightarrow{O_i} + \overrightarrow{Q_i} - 2\overrightarrow{P_i})}{4} - \gamma' \frac{(\overrightarrow{P_{i+1}} - \overrightarrow{P_{i-1}} - \overrightarrow{Q_{i+1}} + \overrightarrow{Q_{i-1}})}{2} - 4\gamma'^2 (\overrightarrow{Q_{i+1}} + \overrightarrow{Q_{i-1}} - 2\overrightarrow{Q_i})$$

$$\vec{b} = \overrightarrow{Q_{i+1}} - \overrightarrow{Q_{i-1}}$$

$$\overrightarrow{S_i} + \vec{a} = \eta' \vec{b} \tag{7}$$

$$\Rightarrow \overrightarrow{S_i} = \eta'\vec{b} - \vec{a}$$

To minimize $\left|\eta'\vec{b} - \vec{a} - \overrightarrow{R_i}\right|$ we have $\eta' = \dfrac{(\vec{a} + \overrightarrow{R_i}) \cdot \vec{b}}{\left|\vec{b}\right|^2}$ (8)

We get $\overrightarrow{S_i}$ by substituting (8) into (7). It is interesting to note that in this method, the extension result is independent of step lengths Δu, Δv and Δs. Having computed $\overrightarrow{R_i}$ and $\overrightarrow{S_i}$ ($i = 1,...n-2$), the two end points are computed by curve extension, either by extending the points on the newly computed slice, or along the corresponding points in the previous slices. For example, to get $\overrightarrow{R_0}$, we fit $\overrightarrow{O_0}$, $\overrightarrow{P_0}$, $\overrightarrow{Q_0}$ into a Bezier curve then extend this curve to plane $z = d$. Figure 7 shows the results on a simple shape.

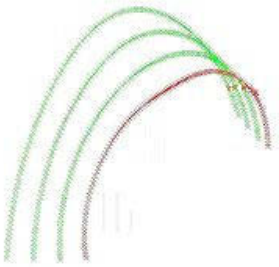

Fig. 7. simple example of extension (red slice)

2.3. Blending morphed and extended slices

The method of section 2.2 extends either the toe geometry, or the back part. Therefore constructing a series of slices from either end will result in a discontinuity when the extended surfaces meet. However, an iterative, weighted propagation scheme can be used to avoid this. The main steps are:

1. Generate one morphed slice at each end of the gap (called MT and MB in figure 8 below)
2. Generate one extended slice at each end of the gap (ET and EB in figure 8)
3. Apply blending function to compute the final slices
   GT = b(z1) MT + (1 – b(z1)) ET, and
   GE = b(z2) MB + (1 – b(z2)) EB,
   where z1, z2 are the Z-coordinates of the planes of slices Mt and EB respectively.
4. Treating the newly generated slices as part of the back/toe, repeat steps 1-3 until the gap is filled.

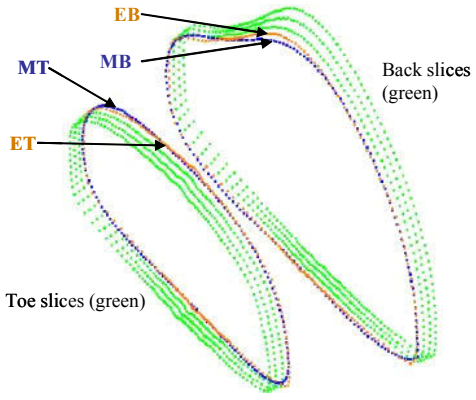

Fig. 8. Morphed and extended slice at each end of gap

Several different types of blending functions, b(z) were tried; figures 9 shows the result of applying a piecewise linear blend, where the slices closer to the boundaries receive most of the weight from extension, and less from the morphing, while slices closer to the center of the gap receive more weight from the morphing. In figure 10, a polynomial blending function is used.

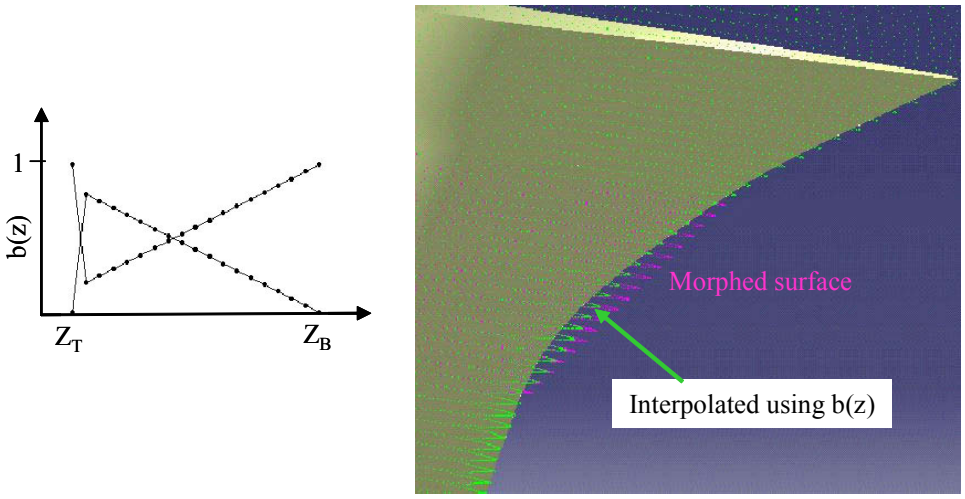

Fig. 9. Linear blending function and resulting interpolated shape

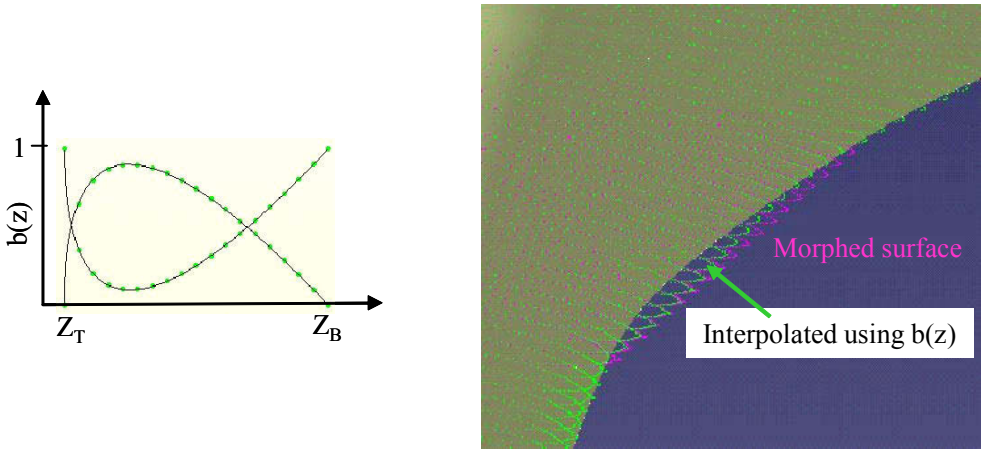

Fig. 10. Polynomial blending function and resulting interpolated shape

2.4. Curvature-change minimizing blends

The above user-defined blending functions gave aesthetic control on the shape, but we may also wish to control fairness. Common curve fairing functionals are based on curvature. If its magnitude is minimized, minimum energy curves (MEC) [17] are formed using $\int \kappa(s)^2 ds$; if its variation is minimized, minimum variation curves (MVC) [17, 18] are created using $\int \kappa(s)'^2 ds$. Similarly, faired minimum energy surfaces (MES) have been created using $\iint_{surface} (\kappa_1^{\,2} + \kappa_2^{\,2}) dA$ [19] and $\iint (\alpha S_{uu}^{\,2} + \beta S_{uv}^{\,2} + \gamma S_{vv}^{\,2}) du dv$ [20, 21]. In the discrete case, the Gaussian curvature, which is concentrated at the corner points of the tessellation used, is used to estimate energy. The corresponding extensions to MVS pose a problem, since directions along which curvature is measured are involved. In

[17], an MVS using principal directions, $\iint_{surface} [(\frac{d\kappa_1}{de_1})^2 + (\frac{d\kappa_2}{de_2})^2] dA$ was introduced. We developed an approximate MVS scheme as follows. A B-spline surface patch is locally interpolated in the neighborhood of a point. We use curvature derivative of curve $C_i(u) = P(u, k_i u)$ (i = 1,2) to approximate $\frac{d\kappa_i}{de_i}$, in which direction $du : dv = k_i$ is a principal direction. The principal directions are derived as the roots of the equation $(FN - MG)k^2 + (EN - LG)k + (EM - LF) = 0$ In which *E, F, G, L, M and N* are the coefficients of first and second fundamental forms. Mapping the two lines, $v = k_i u$ onto the interpolated surface $\boldsymbol{P}$, we get curves $\boldsymbol{P}(u, k_i u) = C_i(u)$. Assuming natural parametric form, the curvature derivative is derived as:

$$\frac{d\kappa_i}{ds} = \frac{d\kappa_i}{du} \cdot \frac{du}{ds} = \frac{d}{du}\left(\frac{|C_i'(u) \times C_i''(u)|}{|C_i'(u)|^3}\right) / |C_i'(u)| =$$

$$\frac{\{[C_i'(u) \cdot C_i''(u)][C_i'(u) \cdot C_i'''(u)]\}|C_i'(u)|^2 - 3|C_i'(u) \times C_i''(u)|^2 [C_i'(u) \cdot C_i''(u)]}{|C_i'(u)|^6 |C_i'(u) \times C_i''(u)|}$$

Using this and the optimization functional over all interpolated points as $\sum (\frac{d\kappa_1}{ds_1})^2 + (\frac{d\kappa_2}{ds_2})^2$, we generate an MVS by a straightforward greedy search algorithm. The algorithm initiates by generating a mesh of points in the gap using linear morphing. It then iteratively modifies coordinates till the above functional is minimized. Figures 11 and 12 show two examples of surfaces generated using this method. For reference, an MES algorithm was also implemented. In figure 11 the green surfaces are the ones being interpolated; the transparent yellow one is the MVS, while the red colored surface is the MES. The first example highlights the differences in the surfaces obtained by these methods. The second example is based on toe and back parts from real shoe last samples.

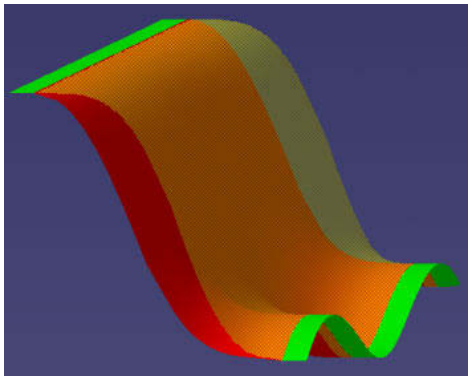

Fig. 11. Example comparing MES and MVS

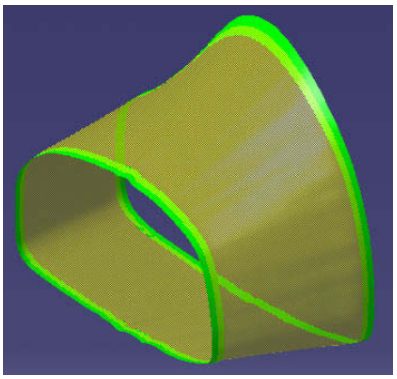

Fig. 12. MVS interpolating back and toe parts from two lasts

3. DISCUSSION

This paper explores some new methods to generate interpolating surfaces between gaps. The application was motivated by a practical problem that arose in the CAD of footwear. The proposed method uses a blending function to interpolate between gap filling surfaces that have been generated using other methods, such as morphing or surface extension. The blending function can be user selected or determined automatically based on higher level constraints placed on the quality of the resulting surface. There are several aspects of the work that remain to be explored. The method presented here works for structured data sets, and it will be useful to extend it to arbitrary point clouds, with gaps that are possibly multi-sided. The energy minimization approach needs to be refined so as to allow more efficient computation. At the same time, the basic approach has proven to yield practically useful outputs in real applications,

and it is hoped that more concrete extensions of the work will result in a useful tool that can be applied to similar problems in other domains.

3. ACKNOWLEDGEMENTS
The authors thank the HK RGC; the research was funded by grant RGC CERG grant # HKUST6235/02E.

4. REFERENCES

[1] Vida, J., Martin, R. R., Varady, T, "A survey of blending methods that use parametric surfaces," Computer-Aided Design, v.26, n5, 341-365, 1994
[2] Peternell, M., Pottmann, H., "Computing rational parametrizations of canal surfaces," Journal of Symbolic Computation, v.23, n.2, 255-266, 1997
[3] Gunter, L., Schicho, J., Winkler, F, Hillgarter, E., "Symbolic parametrization of pipe and canal surfaces," Proceedings of ISSAC '00, St. Andrews, Scotland, 2000
[4] Kim, K. J., Lee, I. K., "Perspective silhouette of canal surfaces," Computer-Aided Design, v.5, n.3, 215-223, 2003
[5] Roach, P. A., Martin, R. R., "Production of blends and fairings by Fourier methods," *Curves and Surfaces in Computer Vision and Graphics III*, J. P. Warren (ed), 162-173, 1992
[6] Moreton, H., Sequin, C., "Surface design with minimum energy networks," *ACM Symposium on Solid Modeling and Applications*, Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications, Austin, TX, USA, 291-301, 1991
[7] Piegl, L, Tiller, W., *The NURBS Book*, Springer-Verlag, Berlin, 1995
[8] Levin A., "Interpolating nets of curves by smooth subdivision surfaces," *Computer Graphics*, Proceedings. SIGGRAPH 99. ACM, New York, NY, USA, 57-64, 1999
[9] Beier, T., Neely, S., "Feature-based image metamorphosis," *Computer Graphics*, Proceedings. SIGGRAPH 92. ACM, v.26, n.2, 35-42, 1992
[10] Lazarus, F., Verroust, A., "Three-dimensional metamorphosis: a survey," The Visual Computer, v.14, n.8-9, 373-389, 1998
[11] Sederberg, T. W., Greenwood, E., "A Physically based approach to 2-D shape blending," *Computer Graphics*, Proceedings. SIGGRAPH 92. ACM, v.26, n.2, 25-34, 1992
[12] W.Boehm, On the definition of geometric continuity, computer-aided design, vol. 20, No.7,1988, pp 370-372.
[13] Thomas Garrity, Joe Warren, Geometric continuity, Computer Aided Geometric Design, 8 (1991), 51-65.
[14] W.Boehm, Curvature continuous curves and surfaces, Computer Aided Geometric Design, 28 (198591), 313-323.
[15] Gary Herron, Techniques for Visual Continuity, *Geometric Modeling: Algorithms and New Trends*, pp163-174, SIAM 1987.
[16] Xue-Zhang Liang, Xiang-Jiu Che, Qiang Li, G2 Continuity Conditions for Two Adjacent B-spline Surfaces, *Proceeding of the Geometric Modeling and Processing* 2004 (GMP'04)
[17] Moreton, H., Sequin, C., Functional Optimization for Fair Surface Design, *Computer Graphics*, v. 26, n. 2, July 1992
[18] Renka, R. J., Constructing fair curves and surfaces with a Sobolev gradient method, *Computer Aided Geometric Design*, 21(2004), pp 137-149
[19] Kallay, M., Constrained Optimization in Surface Design, *Modeling in Computer Graphics*, B.Falcidieno and T.L.Kunii, Eds., Springer-Verlag, 1993, pp 85-93
[20] Hagen, H., Schulze, G., Automatic Smoothing with Geometric Surface Patches, *Computer Aided Geometric Design*, 4 (1987), pp 231-235
[21] Kallay, M., Ravani, B., Optimal Twist Vectors as a Tool for Interpolating a Network of Curves with a Minimum Energy Surface, *Computer Aided Geometric Design*, 7 (1990), pp 465-473