

Styling Evolution for Tight-Fitting Garments

Tsz-Ho Kwok, Yan-Qiu Zhang, Charlie C.L. Wang, *Senior Member, IEEE*,
Yong-Jin Liu, *Member, IEEE*, and Kai Tang

Abstract—We present an evolution method for designing the styling curves of garments. The procedure of evolution is driven by aesthetics-inspired scores to evaluate the quality of styling designs, where the aesthetic considerations are represented in the form of streamlines on human bodies. A dual representation is introduced in our platform to process the styling curves of designs, based on which robust methods for realizing the operations of evolution are developed. Starting from a given set of styling designs on human bodies, we demonstrate the effectiveness of set evolution inspired by aesthetic factors. The evolution is adaptive to the change of aesthetic inspirations. By this adaptation, our platform can automatically generate new designs fulfilling the demands of variations in different human bodies and poses.

Index Terms—styling evolution, aesthetics inspired, design automation, tight-fitting, garment

1 INTRODUCTION

TIGHT-fitting garments are widely used in sports and many medical applications. There are different types of such garments, for example wetsuits, rash-guards, unitards and recovery tights. Most of them are important to the protection of human bodies. Though tight-fitting garments are made to be functional, they are also a kind of fashion. Similar to other fashions, clients would prefer customized sets for themselves.

Customers are not satisfied with the variations produced by using only different colors, fabrics and textures. They wish to have variations in styling designs – in the form of styling curves. At the same time, to achieve a perfect fitting on the final products, the patterns for such garments should be drawn on the 3D surface of human bodies. After that, the surface is decomposed into smaller panels to be flattened into 2D patterns. The final suit can be fabricated by stitching/sewing 2D pieces edge-to-edge into the 3D designed shape. In summary, the function of styling curves for tight-fitting garments is two-fold:

- Functionally, the styling curves are placed where the material needs to be cut such that the whole surface of a human body can be decomposed into a set of flattenable or nearly flattenable panels.

- T.-H. Kwok is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong and the Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology.
- C.C.L. Wang is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong.
- Y.-Q. Zhang and Y.-J. Liu are with the TNLList, Department of Computer Science and Technology, Tsinghua University, Beijing, China.
- K. Tang is with the Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology.
- Co-Corresponding Authors: C.C.L. Wang and Y.-J. Liu (E-mails: cwang@mae.cuhk.edu.hk; liuyongjin@tsinghua.edu.cn).

- Visually, curves are the major styling for tight-fitting garments – different curves give different appearances (see Fig.1 for an example).

Prior research works (e.g., [1]–[4]) focus more on the factor of functionality with emphasis on the ability to be flattened with minimal distortion. The aesthetic factor in the styling design mainly relies on the expertise of designers.

Some companies have provided software to allow designers to draw styling curves interactively on 3D human models and generate the corresponding 2D patterns (e.g., [5]). However, designing a new set of styling curves on human bodies is not only a professionally demanding job but also very time-consuming – it usually takes 3-5 hours for a well-trained designer. Moreover, based on the psychological study, a designer tends to work out designs which are very similar to existing styles after designing a few sets of styling curves in a short period. The motivation of this work is to develop a design evolution system, which starts from limited inputs (e.g., 10 sets in our paper) to generate new styling designs. By using our system, designers only have to provide some high-level intention (in the form of streamlines), and the system can generate a number of outputs based on this intention. With the help of these outputs, designers can quickly generate new styling designs by applying some modifications instead of starting from scratch in every design. Moreover, tight-fitting garments are relatively easy to manufacture. We focus more on aesthetics than manufacturability here.

In this work, we adopt a dual representation of styling curves – that is, each network of styling curves is also formulated as a scalar field. With the help of scalar fields, operators for the topology variation can be realized effectively and efficiently. Robust tools are developed for the conversion between the network of styling curves and its corresponding scalar field.



Fig. 1: Tight-fitting garments in different styles can be generated automatically by our approach. The evolution of styling designs is inspired by aesthetic factors.

One of the most challenging issues in developing an evolution procedure is how to evaluate the quality of an exemplar. In [6], a combination of objective and subjective scores has been employed. Here, we wish the evolution process in our approach to be driven automatically. This paper presents an evolution approach to automatically generate new styling curves with variational topology for tight-fitting garments (see Fig.1). By introducing a computation domain around human bodies, this approach can be generalized to other, normal clothes (see Fig.16).

Our technical contributions are as follows:

- To the best of our knowledge, this is the first approach that tackles the problem of the design evolution of curve networks according to user specified streamlines. Unlike the prior work of part-based model evolution which mainly re-organizes the assembly of components, we present a general curve-based evolution framework which focuses on re-partitioning a two-manifold surface. The evolution approach itself is general although the styling evolution of tight-fitting garments is used to provide example cases in the paper.
- A new method is investigated for generating the fitness score for a network of curves according to aesthetic factors, which are represented by streamlines on manifold surfaces. The quantitative score is statistically estimated by checking the angles between a network of curves and the streamlines. This introduces a new way to allow users to specify their inspirations by aesthetic input (e.g., sketches).
- Our evolution approach relies to a significant extent on the robust implementation of operators for the topological change of curve networks. A dual-representation method is developed in the paper to help realize the operators robustly and compactly. They can be used in more general curve-network evolutions as well.



Fig. 2: Examples of professional design from industry used as the input of our approach. In each pair – (left) the 3D design and (right) the real product.

In summary, we integrate these contributions into a novel evolution framework for generating new designs of styling curves automatically. Tight-fitting garments are employed as examples in the paper (see Fig.2 for one of the professional designs from industry), and we also extend the approach to more general clothes by introducing an additional computation domain around human bodies.

The rest of our paper is organized as follows. After reviewing the related work in Section 2, we present the styling evolution approach in Section 3, which covers the algorithm overview (in Section 3.1), the aesthetics-inspired score evaluation (in Section 3.2) and the dual-representation of styling curves (in Section 3.3). Detail operators employed in the evolution are introduced in Section 4. Lastly, the functionality of our styling evolution approach is demonstrated by a variety of examples in Section 5.

2 RELATED WORKS

2.1 Example-based modeling

Given a database of 3D models, example-based approaches can generate new 3D shapes by compounding parts of shapes inside the database. In the pioneer

work of Funkhouser *et al.* [7], new objects are produced by part-based combination and substitution. This is followed by a thread of example-based modeling work [6], [8]–[12]. Chaudhuri and Koltun [8] developed a synthesis based method to create a new model from input simple models, where the input can be considered as an inspiration of the synthesis. A method was developed in [9] to generate the continuous variability in a collection of 3D shapes without correspondences. A hierarchy-based correspondence is employed in [11] to interpolate between two models with incompatible (different numbers of) components. Kalogerakis *et al.* [10] presented a generative model of part-based shape structure that learns a probabilistic distribution about object styles, part cardinalities, part shapes, and part placements. The probabilistic model is used to supervise the shape synthesis in the domain represented by a collection of example models. Kim *et al.* [12] used an initial template to analyze the collection of models by combining the steps of segmentation, deformation and correspondence together. As a result, their method can work on unstructured and unlabelled data. Xu *et al.* [6] emphasized the idea of *fit and diverse* in using part-based models to evolve novel models.

Our research aims at extending this idea of example based modeling to the field of fashion design. However, the existing part-based evolution techniques cannot be directly applied here as tight-fitting garments have very strict requirements on the compatibility of components' boundaries. Curves instead of parts are more appropriate primitives to be employed in the evolution. There are prior works in the literature involving the evolution of curve networks in computer graphics [13], [14], in which the curve networks are edited directly by using evolutionary techniques. Here the curves must be embedded in two-manifold surfaces (e.g., human bodies). Using graph-based representation and then projecting graphs onto a two-manifold mesh surface can lead to many numerical robustness issues (e.g., the predicates based on approximate arithmetic). New techniques based on the novel dual-representation approach for curve based evolution are presented in this paper.

2.2 Shape blending

The problem of blending the shape of two models has been studied for a long time. Mesh-based approaches (e.g., [15], [16]) require the models to be blended to have the same connectivity. This can be obtained by either cross-parameterization of meshes [17]–[19] when the models have the same topology, or volumetric parameterization [20] when the genus numbers of models are not compatible. In both cases, the correspondences between the structures of models are employed as anchors to govern the blending. The co-segmentation approaches (e.g., [21]–[23]) always serve for this purpose.

To handle the blending between models with different topologies, compact approaches based on implicit functions have been developed. Cohen-Or *et al.* [24] used the signed distance function (SDF) to help generate the metamorphism between 3D models. Turk and O'Brien [25], [26] proposed a shape blending approach based on scattered data interpolation. Specifically, the constraints of two 3D objects are embedded in two planes in a 4D space, and a single implicit function is constructed to interpolate the constraints by using *radial basis functions* (RBF). The intermediate surfaces of blending can be determined by the zero level-set between them. Some followers employed a variant of this approach to fuse components into a watertight model [27]. However, there is no direct solution of using the scattered data interpolation technique for the variation of curve networks.

The work of Stănculescu *et al.* [28] provides a way to represent nested structures (and therefore also their boundary curves) by implicit surfaces, where topology variations can be easily handled by implicit representations. In this paper, we develop a dual-representation (explicit/implicit) for styling curves to realize the evolution operators robustly.

2.3 Garment and toy modeling

Garment modeling is traditionally undertaken by designing 2D patterns, but clothing a 3D character with a garment constructed from 2D patterns is not an intuitive task. Recent work in [29] tries to speed up this procedure of assembly. However, the design process is still time-consuming. Prior work focuses on providing more intuitive and efficient modeling tools to ease the job of designers. Turquin *et al.* [30] generated 3D garments by sketching 2D silhouettes on the model's front and back views. Wang *et al.* [31] developed a platform for interactively designing tight-fitting garments on human bodies by drawing styling curves on 3D human bodies. The curves will later be used to cut the surface into panels which are then unfolded into 2D patterns for use in production. Umetani *et al.* [1] presented a design sensitivity analysis to keep the 2D and 3D perspectives synchronized, and thus artists can interactively edit 2D patterns and see the 3D wrapping results immediately. Yuksel *et al.* [32] proposed a method for interactively modeling yarn-level clothes on 3D human bodies. Fabrication constraints are considered in [33] and partitions are optimized.

On the other hand, some approaches concentrate on automatically adjusting the design of clothes according to the variation of human shapes. DRAPE [34] parameterizes the clothes, with human shapes and poses, in a dataset, such that users do not need to determine a proper size and position of cloth pieces in modeling garments. The research undertaken in [35], [36] automatically generates new garments for

individuals while preserving the original design intention. However, none of the aforementioned work can generate new styles of garments, which is what motivates our work.

Besides garments, some prior works focus on partitioning a given 3D toy model into panels that can be fabricated from 2D pieces of material, where the partition is mainly undertaken in an interactive way [31], [37] or according to geometric factors [2], [3], [38], [39]. Recently, Skouras et al. [33] presented an approach to automatically generate an optimal partition of a model for designing inflatable structures. In contrast, we focus on the evolution of partitions driven by aesthetic factors.

3 STYLING EVOLUTION

In this section, we first brief the overview algorithm of our styling evolution approach. After that, two important aspects, the aesthetics inspired fitness scoring and the dual representation of styling curves, are presented in detail. Note that, in the rest of this paper, we focus on the evolution of styling curves on the same human body. The transformation of design from one human body to another (or the same body with different pose) can be easily obtained with the help of cross-parameterization techniques, which is not the focus of this work.

3.1 Algorithm overview

Our algorithm is developed based on the principles of inheritance, mutation, selection, and crossover from evolution theory. However, the purpose of evolution in our approach is not to search for the ‘best’ of samples. Instead, it serves as a mechanism to drive the generation of new and reasonable styling designs. Based on this reason, we do not conduct a standard evolution algorithm. To have enough new elements in the evolution, a large group of ‘non-best’ samples are retained in the population to increase the diversity.

Given a set of exemplar styling designs $S^0 = \{s_i\}$ ($i = 1, \dots, n$) serving as the current population, our algorithm goes through the following steps to produce the next generation of designs.

Step 1): First, $5n$ pairs of samples are randomly selected from S^g , where the *crossover* operator is applied to each pair of samples, $\langle s_i, s_j \rangle$, to generate a new sample s_k . In total, $5n$ new samples are generated. Note that, the crossover of two styling designs is also driven by a random number generator (details can be found in Section 4.1). According to the feedbacks in our statistical study [40], users always try to avoid a design with too many patches – they feel *simple is beautiful*. They give very low rating to those designs with more than 50 patches. To incorporate this factor in our evolution framework, we discard any result of crossover when it contains more than 50 patches.

Step 2): Among the set of newly generated samples, the *mutation* operator (Section 4.2) is applied to $P_m\%$ of all the samples. After that, $P_s\%$ of samples are converted to a symmetric design by the *symmetrization* operator (Section 4.3). All the samples are randomly selected¹.

Step 3): The fitness score is then evaluated on each of these $5n$ samples. The score is inspired by the aesthetic factors specified on the human bodies by sketching input (detail of the evaluation method for fitness score is presented in Section 3.2)

Step 4): n samples are selected to survive. Here we have two options for using the evolution framework for different purposes, where the *optimal inspiration* scheme is employed to improve the quality of styling curves and the *general inspiration* scheme is used to increase the diversity of exemplars.

- *Optimal Inspiration:* When this scheme is applied, 80% of the retained n samples are selected from the ‘best’ designs with their fitness scores higher than others. To keep a certain diversity of samples in the next generation, 20% samples are randomly selected from the rest. The set of designs in the next generation, S^{g+1} , is then formed by this selection step.
- *General Inspiration:* The main purpose of this strategy is to increase the variants of styling designs. All the $5n$ samples are sorted according to their fitness scores and every $5n/(n-3)$ samples are grouped into a bin. In total, $(n-3)$ bins are formed. One sample is randomly selected from each bin to enter the evolution in the next generation. The samples selected in this way are uniformly distributed in different ranges of the fitness scores. As a result, enough diversity has been retained for the evolution. In order to keep the features on the input designs, three samples are randomly selected from S^0 (i.e., the input set) to add into S^{g+1} .

Iteratively running the above steps, the algorithm evolves the set of styling designs. Note that a robust implementation of the operators in evolution is crucial to the success of this algorithm, which will be detailed in Section 3.3. The values of parameters used in different steps and operators of the algorithm are listed in Table 1.

3.2 Fitness score

Fashion is a kind of unpredictable ‘magic’. It is hard to quantitatively measure how ‘good’ a design is. Therefore, our framework provides designers an interface to specify their preferences in terms of streamlines on human bodies, which is considered as an aesthetics inspiration given by designers. The platform then

1. In our implementation, satisfactory results can be obtained when using $P_m = 10$ and $P_s = 30$.

TABLE 1: Values of Parameters

	Variable	Meaning	Value
Evolution	n	population	10
Evolution	$P_m\%$	% for mutation	10
Evolution	$P_s\%$	% for symmetrization	30
Evolution	-	optimal inspiration	80 : 20
Fitness	m	histogram intervals	10
Dual Rep.	δ_d	field offset	1cm
Dual Rep.	d_{max}	scaled max. field	5
Dual Rep.	d_{min}	scaled min. field	$2d_{max}$
Crossover	w_v	confidence coefficient	0.0 – 1.0
Crossover	-	buffer region	5 – 15cm
Mutation	α	random no.	0.5 – 1.0
Mutation	D_s	width of shoulder	by model

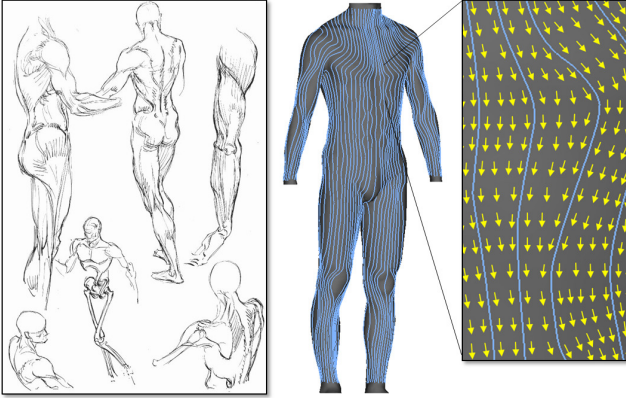


Fig. 3: (Left) The picture comes from Loomis’s drawing book about the male’s muscle lines, with the reference to which the streamlines are drawn on the 3D human bodies (right). The fitness score in our framework is evaluated according to how much the styling curves follows the aesthetics inspired streamlines.

measures how much do the styling curves of a design follow this preference.

Streamlines: The streamlines are defined on the surface H of a human body as a vector field – i.e., for any point \mathbf{p} on H , a vector $\mathbf{s}(\mathbf{p}) \in \mathbb{R}^3$ is defined. In practice, we represent $\mathbf{s}(\mathbf{p})$ by a piecewise-linear approximation on H , $\tilde{\mathbf{s}}(f)$. That is, the vector in any triangle $f \in H$ is a constant. After designing some ‘anchor’ streamlines (with a sketching tool) on the surface H , the vectors in other triangles can be determined via a scattered data interpolation technique (ref. [41]). The streamline starting from a point $\mathbf{p} \in H$ can be obtained by tracing along the vector field $\tilde{\mathbf{s}}(f)$ as shown in the right of Fig.3. Note that, when evaluating the fitness score, we do not need to have explicit representation of the streamlines. The evaluation only involves the vector field $\tilde{\mathbf{s}}(f)$. One example of streamlines can be defined according to the anatomy study of bones and muscles. As shown in Fig.3, the streamlines are defined with reference to Loomis’s drawing book [42]. Designers can adjust (or even redefine) the streamlines by their own preferences to drive the evolution of styling designs.

Angular score: One of the dual representation of

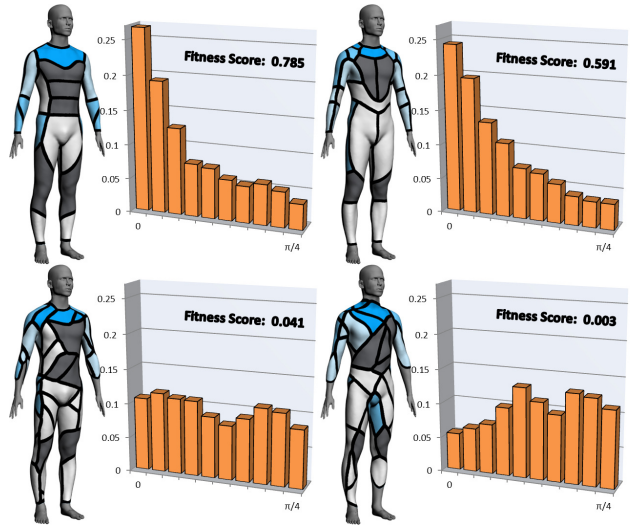


Fig. 4: Fitness scores evaluated on different examples w.r.t. the streamlines shown in Fig.3. The fitness score is computed from a histogram indicating the distribution of angles between the styling curves and the streamlines.

styling curves is a set of piecewise linear curves lying on the surface H of a human body. Considering a segment l of styling curve falling inside a triangular face $f \in H$, the angle θ between this portion of the styling curve and the streamlines is evaluated by $\theta = \arcsin\left(\frac{\|l \times \tilde{\mathbf{s}}(f)\|}{\|l\| \|\tilde{\mathbf{s}}(f)\|}\right)$. The feedback we obtained from industry finds that the design of styling curves looks more aesthetic when they are nearly parallel or perpendicular to the streamlines. Therefore, we score each linear segment l on a styling curve as

$$E(l) = \frac{\pi}{4} - \left| \frac{\pi}{4} - \theta \right| \quad (1)$$

by its corresponding angle θ to the streamlines. $E(l) \in [0, \frac{\pi}{4}]$.

Fitness function: Although it is not easy to quantify how good a design is good, the fitness score evaluated as follows can clearly distinguish the ‘bad’ ones. Histogram for the distribution of angular scores on styling curves is constructed for this purpose. First, we evaluate the angular score on each line segment of styling curves. To build the histogram, the interval $[0, \frac{\pi}{4}]$ is segmented into m sub-intervals with equal size. $m = 10$ is adopted in our implementation. For a line segment l if its score $E(l)$ falls in the i -th interval, the histogram of that interval, h_i , is increased by $\|l\|/L$ where $\|\cdot\|$ denotes the magnitude of a vector and L is the total length of all the styling curves on a design. By this way, we can obtain a histogram that indicates the distribution of angles between the styling curves and the streamlines. Some examples are shown in Fig.4. Note that the histogram has a common property that $\sum_i h_i \equiv 1$. When the styling curves of a design follow the streamlines, the histogram is mainly located at the left side and drops promptly. We therefore formulate the fitness score as a Gaussian weighted integration

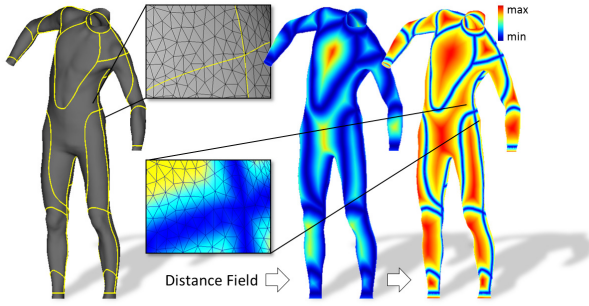


Fig. 5: The piecewise-linear styling curves are converted into a design-field (most right) with the help of a geodesic distance-field (middle).

of the dropping speed (i.e., $(h_{i-1} - h_i)$).

$$F = m \sum_{i=1}^{m-1} \frac{2}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{(i-1) - \mu}{\sigma} \right)^2} |h_{i-1} - h_i|_+, \quad (2)$$

where we use $\sigma = \frac{1}{6}(m-1)$ for the standard derivation of the Gaussian function, the mean μ is set to zero and $|\cdot|_+$ returns *zero* for negative values. The negative difference are truncated to zero so that a ‘poor’ design with an increased value of h_i (while the increasing of i) will gain *zero* on this score. The fitness score evaluated by Eq.(2) returns a quantitative measurement of styling curves, the larger the better. Four examples can be found in Fig.4. A survey has also been taken to validate our formulation of the fitness score – details can be found in Section 5.

3.3 Dual representation

A dual representation is employed for the styling curves so that the operators of evolution can be realized robustly. A set of styling curves can be explicitly stored as piecewise-linear curves, where each line segment of a curve is coupled with a triangular face on the surface of a human body. The vertices of line segments are positioned by their barycentric coordinates in the triangles. The styling curves can be automatically transferred when the shape and pose of human bodies are changed. Another representation of the styling curves is a piecewise-linear scalar field with its field values stored at the vertices of a human body’s mesh surface. This field is named as the *design-field*. The set of points with negative field-values form an implicit representation of the banded region around styling curves. We now discuss the conversion between these dual representations.

Curves-to-field: The design-field is constructed with the help of a geodesic distance-field. For the given surface H of a human model, the geodesic distance from every vertex on H to the line segments of the styling curves is first computed by the method of [?], [43]. In order to ease the job of curve reconstruction from the design-field, the field values on all vertices are reduced by δ_d to create a banded region with

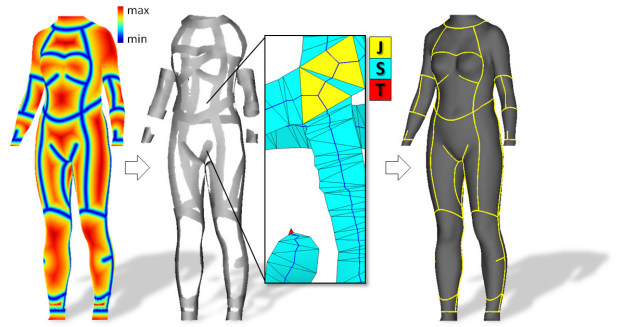


Fig. 6: A design-field can be converted back into the piecewise-linear styling curves with the help of chordal axis extracted from the triangulation of a banded region. ‘J’, ‘S’ and ‘T’ stand for junction, sleeve and terminal triangles respectively.

negative field-values. We set $\delta_d = 1cm$ for models in the real size of a human body. The banded regions segment the whole surface H into separated patches. Unlike a distance-field, each patch surrounded by styling curves in a design-field should present similar importance during the evolution. For this reason, the field values within a patch are scaled to a fixed range $[-d_{min}, d_{max}]$. Assume g_{max} is the maximal value in the geodesic distance field of a particular patch, the design-field in the region of this patch is updated by scaling the positive values with d_{max}/g_{max} and scaling negative ones with d_{min}/δ_d . In this way, the whole design-field is re-scaled patch by patch. An example of the curves-to-field conversion is shown in Fig.5.

Remarks. The range of values in a design-field is an important issue when merging two design-fields (as what is processed in the crossover operation in Section 4.1). If $d_{max} \gg d_{min}$, blending two design-fields by the equal weights (i.e., computing the average) can discard almost all styling curves. In this case, troughs of the design-field can be easily elevated to above the zero level-set. Therefore, we should reverse the relationship by letting $d_{min} > d_{max}$. According to our experimental tests, letting $d_{min} = 2d_{max}$ performs well to avoid the above problem.

Field-to-curves: Given a design-field, $d(\mathbf{p})$ ($\forall \mathbf{p} \in H$), a robust algorithm is developed to extract the piecewise-linear curves from the banded regions with negative field values. First of all, the triangles overlapped with the banded regions are selected out from H to construct a new mesh surface B . Then, a mesh simplification procedure [44] is applied to B until only the vertices on the boundary of B are retained. Note that, the vertices that are originally located at the boundary should be kept unchanged during the simplification. After that, the resultant mesh surface B' is transformed to chordal axis that was used in [45]. Specifically, the triangles on B' are divided into three types: *terminal* triangle – a triangle with two edges on the boundary, *sleeve* triangle – a triangle with only one

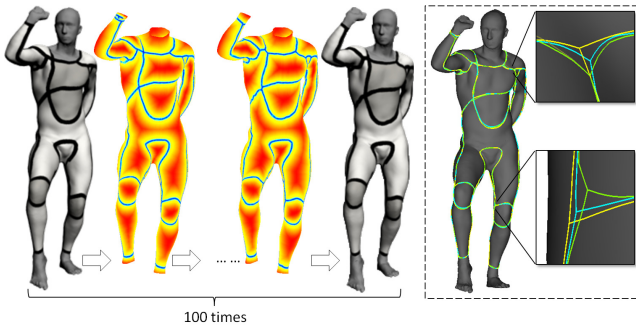


Fig. 7: An example to test the accuracy of conversion between styling curves and the design-field, where the conversion has been repeatedly applied for 100 times. In the most right figure, the curves (displayed in different colors) are extracted after the 1st (in pink hidden by the original curves in yellow), the 50th (in blue) and the 100th (in green) conversions and are superimposed together to check the change of curves.

edge on the boundary, and *junction* triangle – a triangle with no edge on the boundary. The chordal axis is obtained by connecting the midpoints of the edges that are adjacent to two triangles and the centers of the terminal and junction triangles. An illustration can be found in the zoom view of Fig.6. To remove the insignificant branches in the resultant chordal axis, the pruning algorithm introduced in [46] is also applied. For a terminal triangle, if it is connected to a junction triangle with very few sleeve triangles (e.g., less than three in our practice), this branch will be removed. Lastly, line segments of the chordal axis are mapped back to the surface H by projecting the endpoints onto the closest face and connecting the projected points by a geodesic curve.

Remarks. The conversion between styling curves and a design-field can be robustly implemented by the above method. In other words, it does not suffer from the problems caused by the inexact numerical predicates. On the other aspect, although it is robust, the conversions between styling curves and the design-field is not lossless. Caused by the approximation error, the styling curves reconstructed from a design-field may be distorted away from the original styling curves that are used to construct the design-field. To see how significant the conversion error is, one can convert the styling curves of a design into a design-field and then convert the design-field back into piecewise-linear curves. After repeatedly performing the conversion 100 times, we find that the overall styling curves are only slightly moved (see Fig.7). The most significant change is the movement of sharp corners, which is still in an acceptable range. Therefore, the conversion between the explicit and the implicit representations of styling curves only introduces small errors.

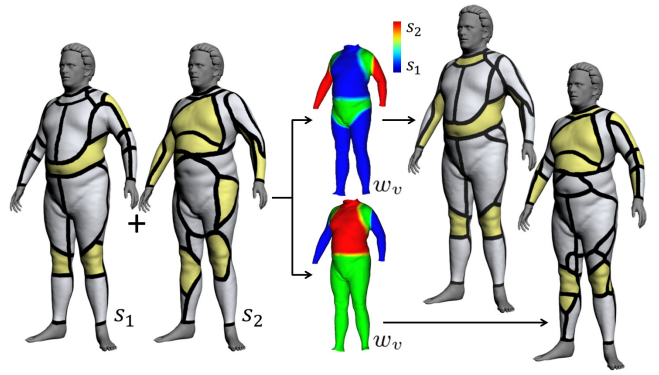


Fig. 8: The crossover operator is realized with the help of confidence coefficients w_v . Different maps of w_v lead to different results.

4 OPERATORS

4.1 Crossover

Given two design-fields, $d_A(\mathbf{p})$ and $d_B(\mathbf{p})$, defined on the surface H of a human body (i.e., $\mathbf{p} \in H$), a crossover operator is demanded to generate a new design-field $d_C(\mathbf{p})$ that inherits features from both design A and design B . Simply blending or superposing $d_A(\mathbf{p})$ and $d_B(\mathbf{p})$ cannot achieve this goal. We borrow the idea from parts-based shape evolution to ‘assemble’ the styling curves in different regions selected from A and B . Unlike the prior work, the styling curves with different topology in two neighboring regions must be merged smoothly. This is actually a topology metamorphosis problem, and we tackle it by introducing buffering regions between different parts.

A hierarchical structure with three levels is constructed on the surface of human body for the parts-based crossover operator. The root is a complete human body. In the second level, this human model is decomposed into arms, upper-body and legs. In the finest level, each of the above three parts are further decomposed into smaller regions together with an option that there will be no further decomposition. Buffering regions are then constructed between parts to absorb the incompatible styling curves selected from different designs. Specifically, the crossover of $d_A(\mathbf{p})$ and $d_B(\mathbf{p})$ is realized with the help of this hierarchical structure as follows.

- First, every vertex v on H is associated with a confidence coefficient w_v for specifying whether the design obtained from crossover at this point comes from A (when $w_v = 0.0$) or from B (by letting $w_v = 1.0$). The values are initially assigned to *undefined*.
- Now we start to take random selection of parts in the second level of hierarchy. k parts are selected for the design A and the rest are selected from B . And in the finest level, the portions (or the whole part) of arms, legs and upper-body are randomly picked. For the vertices in the regions selected from A , their confidence coefficients are assigned

as $w_v = 0.0$, and $w_v = 1.0$ are assigned to those regions selected from B . For each selected part, a buffering region with a random width varying between 5 – 15cm are introduced near the boundary by retaining the *undefined* confidence coefficients. Note that, it is possible to have unselected regions – e.g., the upper-arm when the forearm is selected from A (or B). For these regions, $w_v = 0.5$ are employed and a buffering region is also added near the boundaries. The styling curves from A and B are superimposed in these regions.

- Lastly, a bounded biharmonic field by enforcing $w_v \in [0, 1]$ is solved on those ‘undefined’ vertices (ref. [47]). The biharmonic field solves the problem of smooth merging. After that, the new design-field is generated by the linear-blending of d_A and d_B with the weights w_v .

Our crossover operator can effectively generate different styling designs by inheriting features from A and B (see Fig.8 for an example). Note that, after generating a new design-field, the values of the design-field must be regularized by going through the procedure of field-to-curves and curves-to-field conversions.

4.2 Mutation

A mutation operator consists of two essential steps: patch-erosion and curve-removal, where the first step is realized on the design-field and the second one is applied to the set of styling curves.

Patch-erosion: First of all, we randomly pick one or two largest patches across the center plane (i.e., the middle plane that separates a human body into left-right parts) of the human body to apply the patch-erosion. To erode one patch P , we first find $d_{\max}^P = \max_{\forall \mathbf{p} \in P} \{d(\mathbf{p})\}$ and then we reduce the field value at every point by

$$d'(\mathbf{p}) = d(\mathbf{p}) - \alpha d_{\max}^P \left(1 - \frac{4D_c(\mathbf{p})}{D_s}\right), \quad (3)$$

where α is a random number $\in [0.5, 1.0]$, D_s is the width of a human body’s shoulder, and $D_c(\cdot)$ returns the geodesic distance from a point to the central vertical trunk on the human body. We can shrink the boundary of P by changing the field value in this way. An example can be found in Fig.9(a).

Curve-removal: The purpose of this mutation step is to remove those curves not following the aesthetical inspirations. The fitness score of a curve C can be computed by

$$E(C) = \frac{1}{L_C} \sum_{\mathbf{l} \in C} \|\mathbf{l}\| E(\mathbf{l}), \quad (4)$$

where \mathbf{l} is a line segment on C , $L_C = \sum \|\mathbf{l}\|$ is the length of C , and $E(\mathbf{l})$ evaluates the score of line segment by Eq.(1). All curves are sorted by their

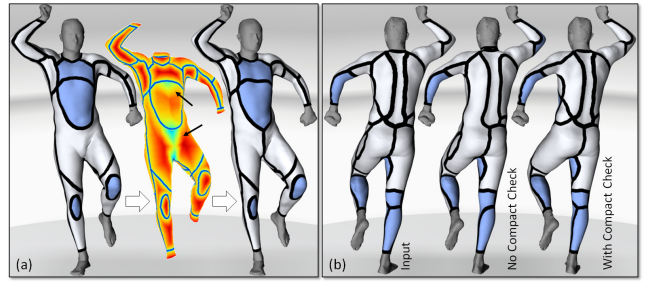


Fig. 9: An illustration of the mutation operator. (a) The patch-erosion step, where the front piece is split into two after applying this erosion. (b) The curve-removal step with vs. without the compactness criterion – it can find that many patches are not compact when the compactness criterion is not used.

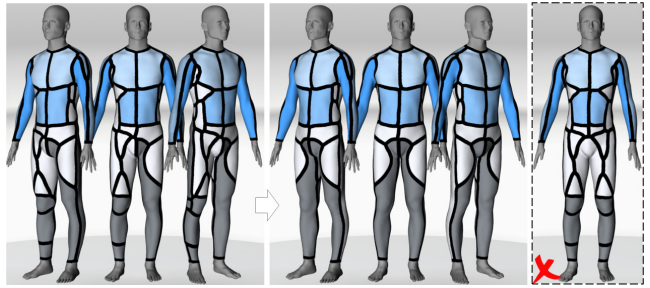


Fig. 10: The symmetrization operator applied to a design of styling lines will retain the half with higher fitness score. In this example, the fitness score is $F = 0.497$ for the input design where $F_{left} = 0.590$ and $F_{right} = 0.415$. The resultant model of symmetrization gives $F = 0.607$. From the most right image, it is easy to find that the result is poor (with $F = 0.317$) when the half with lower score is retained and duplicated.

fitness scores in descending order. We remove the curves with ‘poor’ fitness scores (i.e., higher values) one by one. During the curve-removal, another factor of aesthetics is also considered – compactness. Many animals and plants in nature are formed by compact components, which is thought as beautiful [48]. Inspired by this study, we incorporate the criterion of compactness into the curve-removal operator. The compactness of a patch \mathcal{P} can be evaluated by $C(\mathcal{P}) = L_{\mathcal{P}}^2 / A_{\mathcal{P}}$ with $L_{\mathcal{P}}$ and $A_{\mathcal{P}}$ respectively denoting the boundary’s length and the area. After removing a curve, its left and right patches, \mathcal{P}_L and \mathcal{P}_R are merged into a new patch \mathcal{P}_{new} . If $C(\mathcal{P}_{new})/C(\mathcal{P}_L)$ or $C(\mathcal{P}_{new})/C(\mathcal{P}_R)$ is greater than 1.5, we will abandon the removal of this curve. Figure 9(b) shows an illustration of the curve-removal with vs. without this compactness criterion. After removing a curve C , the curves originally adjacent to C will be stretched on the surface to avoid sharp corners. The removal of curves stops when the summed length of all the removed curves has been more than 5% of the total length of styling curves.

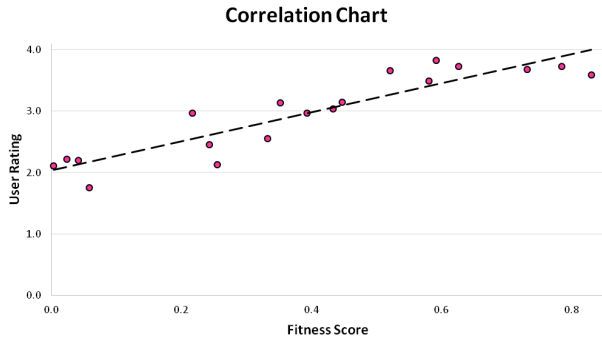


Fig. 11: A survey to validate our formulation for the fitness score of styling curves: The fitness scores evaluated by Eq.(2) are compared with the statistical ratings generated by 172 individuals. It can be found that the fitness score and the user rating are highly correlated with the correlation coefficient as 0.9.

4.3 Symmetrization

Although the styling curves of a good design is not necessary to be symmetric, the perception of human beings always prefers symmetric structures. To facilitate the operation of symmetrization, we need to construct the point-to-point mapping between left and right bodies on a human model. A surface model H is first mirror-copied to an inverse model, H' , along the horizontal axis. As every triangle on the left part of H' is obtained from a triangle on the right of H , the left-to-right mapping (and vice versa) on H can be obtained by the cross-parameterization between H and H' . With the help of this left-right mapping, we can easily assign the field values of vertices on the left of a human body by the design-field at the right, and vice versa. Before applying the symmetrization operator to a design, we first use the fitness score in Eq.(2) to check the styling curves on the left versus the right. The side with higher score is selected to copy to the other side. As illustrated in Fig.10, this operation can always improve the fitness score.

5 RESULTS AND DISCUSSION

This section presents test results of our styling evolution. Statistical studies are taken to verify the effectiveness of the fitness scores developed in this work. The styling evolution approach has been tried in a variety of experimental tests. The framework presented in the paper has been proved to work successfully in both the general inspiration and the optimal inspiration.

5.1 Validation of fitness score

The fitness score formulated in Eq.(2) is used in our framework of styling evolution to find out those 'poor' designs. To validate its soundness and effectiveness, we have taken a statistical study based on questionnaires [40]. This study includes 20 different designs of styling curves with their fitness scores

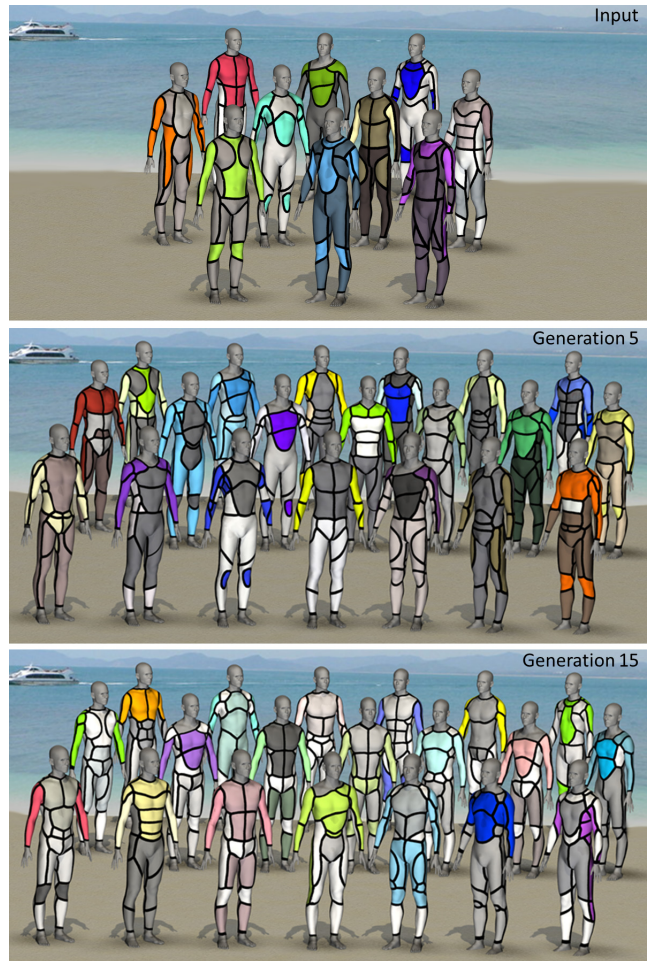


Fig. 12: Evolution of a wetsuit set for male models, the streamlines of which is defined as shown in Fig.3. More details can be found in accompany demo video.

varying from 1.25 to 0.00313. Each design is provided with a front-view, a back-view, and an isometric-view in the questionnaires. Each design is evaluated with a 5-scale rating, where 5 stands for the best and 1 denotes the poorest. 172 subjects (103 males and 69 females) mainly aged between 20 – 30 took part in the survey. 34% of them have design experiences, 35% have little experiences and the rest are novices in design. The rating results are shown in Fig.11 to compare with the fitness scores obtained by Eq.(2). The styling designs are sorted by their fitness scores in descending order. Generally speaking, the trend of ratings given by individuals follows the fitness scores but with some perturbation. It can be found that, for a design with very low rating, its fitness score is also very low. Therefore, when using this formulation of fitness scores in the evolution procedure, the 'poor' designs will be eliminated in the step of selecting samples for the next generation.

5.2 General inspiration

The general inspiration scheme of our framework is to generate diversified designs. Given a set of styling

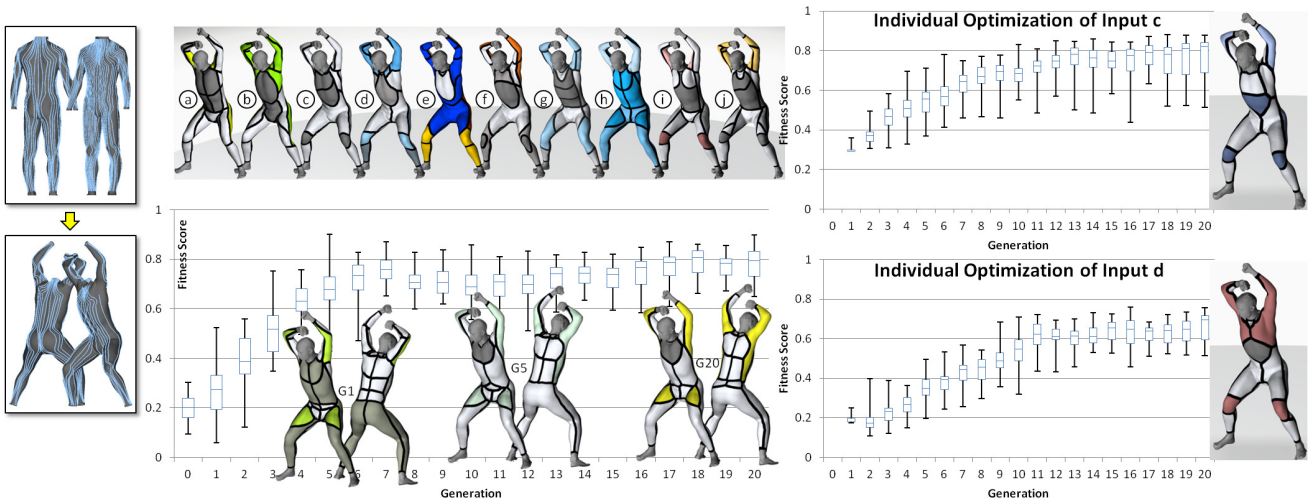


Fig. 14: Evolution of a wetsuit set for male models in a different pose with different streamlines. After changing the streamlines by designers according to the new pose, the evolution of styling curves is affected by the changed styling curves. (Left-bottom) The evolution by taking 10 design from ‘a’ to ‘j’ as input. (Right-top) The optimal inspiration by using styling ‘c’ and its variants generated by mutations as input and (right-bottom) the optimal inspiration from ‘d’ and its variants. It can be found that styling curves have been optimized to not pass through the region of arm-hole. These results follow the intension of designers that no streamline is given at the arm-hole.

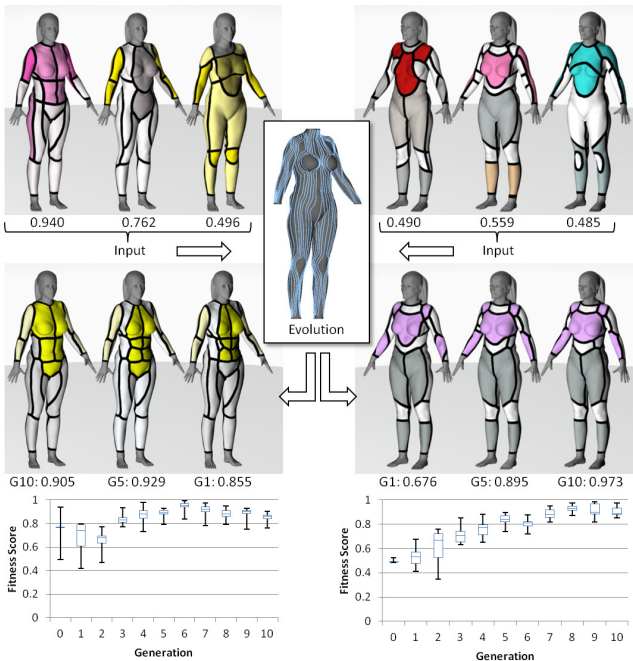


Fig. 13: Evolution for optimal inspiration: given a set of styling designs that are originally for male models, the framework can evolve the styling curves into a design fulfill the features (specified by streamlines) of female models.

designs for male’s wetsuit, the framework can generate a large amount of variations after running a few generations. Examples are shown in Fig.12. A set of styling curves generated by this approach can then be applied as input for the design software (e.g., the CAD system [5]) to generate 2D patterns. With the help of numerically controlled cutting machines, the

life cycle of design and manufacturing for customized tight-fitting garments can be tremendously reduced.

5.3 Optimal inspiration

When the scheme of optimal inspiration is adopted in the evolution framework, this approach will help improve the quality of styling curves (i.e., to drive the curves to follow the trend dictated by streamlines). In the example shown in Fig.13, the input sets are originally designed for the male models. Our evolution framework can optimize the styling curves to let them follow the features of female bodies.

Different styling designs for females can be obtained by taking different exemplar sets as input to the evolution framework (see the right of Fig.13). The convergence of the evolution can be studied through the change of fitness scores in different generations of exemplars (i.e., $\forall s_i \in \mathcal{S}^g, F(s_i)$). In general, the fitness score of the ‘best’ sample,

$$F_{best}^g = \max_{s_i \in \mathcal{S}^g} \{F(s_i)\}, \quad (5)$$

increases during the evolution (although not monotonically).

In nature, the shape of muscles on a human body changes when it is in different poses (especially in the region with high curvature). As a result, the streamlines obtained from Langer’s lines [49] have different distributions for the same person in different poses. The distribution of streamlines in a different pose cannot be automatically obtained from an isometric (or conformal) deformation of human bodies. Our system requires designers to interactively modify the streamlines after changing a human body’s pose.

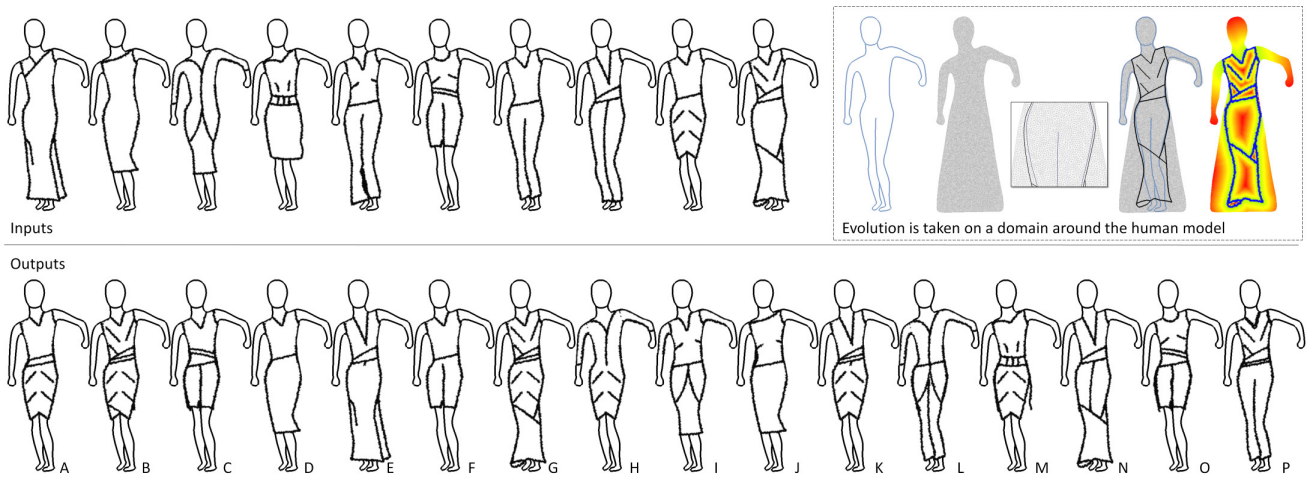


Fig. 16: Evolution of styling designs for general clothes: the evolution of styling curves are taken on a mesh enclosing the human model. Given the input styling designs as shown in the first row, more styling designs can be inspired from input by the evolution framework. Again, the dual representation is employed for the styling curves.

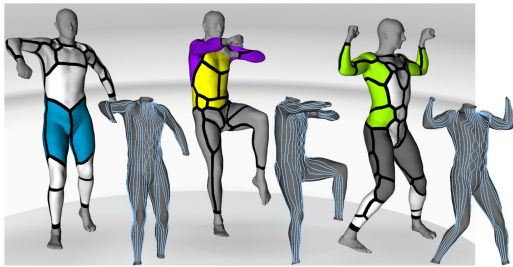


Fig. 15: Examples of designs generated by different streamlines used for different poses, which are obtained from the SCAPE database [16].

Then, the evolution of styling curves is controlled by the aesthetic input – the streamlines defined on the surface of human models. Each specific sport will have a common pose of human models in that sport (e.g., surfing and diving should have different common poses). Given the streamlines from designers for a particular pose, the styling designs generated by our system will follow the streamlines for that pose to make user feel comfortable (e.g., no seam line at the arm-hole, etc.). Figure 14 gives the evolution results when the streamlines of human models are changed according to the change of poses. Here, the poses of human bodies are obtained from the SCAPE database [16]. Results from different poses and their corresponding streamlines are shown in Fig.15.

Our framework actually provides a tool for designers to add their influence on the procedure of the styling evolution by using a sketching interface (e.g., [41]) to produce the vector fields of streamlines on human models. Change the distribution of streamlines can drive the evolution of designs moving to different styles.

5.4 Application on general fitting clothes

We have extended the styling evolution technique in the paper to process more general styles of garments (e.g., pants, dresses and skirts shown in Fig.16). Without loss of generality, we can construct a computational domain (in the form of a mesh surface) around and enclosing human bodies. The styling evolution for general clothes is then performed on this computational domain instead of the human bodies. The same evolution framework developed in this paper can be applied. In our current implementation shown in Fig.16, only crossover operators have been applied for the general fitting clothes. The mutation operators for general clothes will be developed in the near future. The current implementation of evolution for general clothes produces styling designs in the form of 2D curves.

This generalization can actually be applied to any products fabricated by partitioning a given manifold surface into patches that can be fabricated from planar materials. As long as the mapping between computation domains can be constructed, the operators developed in the paper for manipulating the network of curves can be applied to realize the evolution of curve-networks on the computation domains.

5.5 Discussions

There are some limitations of our approach.

Streamlines: First of all, although the fitness function is based on streamlines, the streamlines do not change automatically according to the deformation of human bodies. Our current strategy follows the demand from industry to specify a set of streamlines for a particular pose. It is time-consuming to specify streamlines on all different poses. One future work is to develop a method to generate the streamlines automatically based on the Langer’s lines [49]. Although streamlines

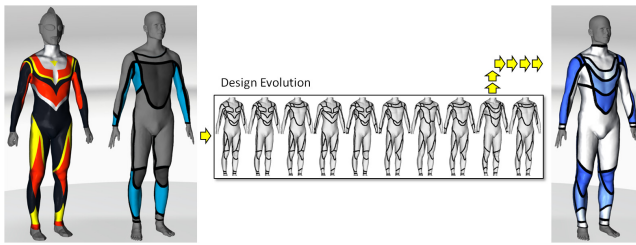


Fig. 17: Using a design that is significantly different from common wetsuits can increase the diversity of evolution.

used in our approach need to integrate the intension of designers, Langer’s lines can be used as a good starting point.

Manufacturing: In our current framework, the constraints of manufacturability (e.g., developability of patches) are not added based on two reasons. First, during the styling design conducted by professional designers in industry, they rarely consider about the developability. Such constraints are mainly considered by another group of people – the pattern designers. Second, for the tight-fitting clothes like wetsuit, the patterns usually are small so they can easily be flattened into 2D meanwhile keeping the boundary length [31]. When the boundary lengths have been kept, there is no difficulty in stitching or sewing the patterns together. Another constraint on the fabrication of tight-fitting garments is the total length of seam lines. The longer, the higher cost is in both the time and the material usage. Moreover, having too many seam lines will make users feel uncomfortable. We have considered the length factor of seam lines in the evolution framework.

Other constraints: The major factor considered in our evolution framework is to let the layout of styling curves following the designer-specified streamlines. Other constraints such as the wearability and the comfortability of users have not been considered in the objective function of design evolution. The major difficulty of integrating them into the current evolution framework comes from the lack of an effective method to evaluate these factors in a quantitative way. Complex physical simulations should be involved.

Diversity: To increase the diversity of designs generated by our evolution framework, we need to introduce example input that are significantly different from the common design of wetsuit. As shown in a preliminary test (see Fig.17), more diverse styling curves can be generated when an cartoon Ultraman design and a wetsuit are used as input. In our future work, more methods for increasing the diversity of evolution will be studied.

6 CONCLUSION

In this paper, we propose an evolution framework for designing the styling curves of garments. The tight-

fitting garments are taken as an example to explore how to create new styling curves on the surface of human bodies. The evolution is driven by a well-defined fitness score, where the aesthetical factors are specified as streamlines. With the help of this tool, a computer system can automatically generate new designs satisfying the demands of variants in different human bodies and poses. A dual representation is proposed in the paper to realize the robust implementation of operations on styling curves including crossover, mutation, and symmetrization. Experimental results shown in the paper have demonstrated the functionality of this approach.

ACKNOWLEDGMENTS

This work is partially supported by the Hong Kong Innovative Technology Fund (ITS/060/13), and the Natural Science Foundation of China (61322206, 61432003, 61272228) and the National Basic Research Program of China (2011CB302202). Support for Tsz-Ho Kwok by an internship from Hong Kong ITF (GHP/057/12) is appreciated.

REFERENCES

- [1] N. Umetani, D. M. Kaufman, T. Igarashi, and E. Grinspun, “Sensitive couture for interactive garment modeling and editing,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 90:1–90:12, 2011.
- [2] P. Decaudin, D. Julius, J. Wither, L. Boissieux, A. Sheffer, and M.-P. Cani, “Virtual garments: A fully geometric approach for clothing design,” *Computer Graphics Forum*, vol. 25, no. 3, pp. 625–634, 2006.
- [3] D. Julius, V. Kraevoy, and A. Sheffer, “D-charts: Quasi-developable mesh segmentation,” *Computer Graphics Forum*, vol. 24, no. 3, pp. 981–990, 2005.
- [4] C. C. L. Wang, Y. Wang, K. Tang, and M. M. F. Yuen, “Reduce the stretch in surface flattening by finding cutting paths to the surface boundary,” *Computer-Aided Design*, vol. 36, no. 8, pp. 665–677, 2004.
- [5] Exuskin, “3D Pattern Design,” <http://exuskin.com/>, 2013.
- [6] K. Xu, H. Zhang, D. Cohen-Or, and B. Chen, “Fit and Diverse: Set evolution for inspiring 3D shape galleries,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 57:1–57:10, 2012.
- [7] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin, “Modeling by example,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 652–663, 2004.
- [8] S. Chaudhuri and V. Koltun, “Data-driven suggestions for creativity support in 3d modeling,” *ACM Trans. Graph.*, vol. 29, no. 6, pp. 183:1–183:10, 2010.
- [9] M. Ovsjanikov, W. Li, L. Guibas, and N. J. Mitra, “Exploration of continuous variability in collections of 3d shapes,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 33:1–33:10, 2011.
- [10] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun, “A probabilistic model for component-based shape synthesis,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 55:1–55:11, 2012.
- [11] A. Jain, T. Thormählen, T. Ritschel, and H.-P. Seidel, “Exploring shape variations by 3D-model decomposition and part-based recombination,” *Comp. Graph. Forum*, vol. 31, no. 2-3, pp. 631–640, 2012.
- [12] V. G. Kim, W. Li, N. J. Mitra, S. Chaudhuri, S. DiVerdi, and T. Funkhouser, “Learning part-based templates from large collections of 3D shapes,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 70:1–70:12, 2013.
- [13] K. Sims, “Artificial evolution for computer graphics,” *SIGGRAPH Comput. Graph.*, vol. 25, no. 4, pp. 319–328, Jul. 1991.
- [14] K. Sims, “Evolving virtual creatures,” in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’94, 1994, pp. 15–22.

- [15] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 399–405, 2004.
- [16] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: Shape completion and animation of people," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 408–416, 2005.
- [17] V. Kraevoy and A. Sheffer, "Cross-parameterization and compatible remeshing of 3d models," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 861–869, 2004.
- [18] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe, "Inter-surface mapping," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 870–877, 2004.
- [19] T.-H. Kwok, Y. Zhang, and C. C. L. Wang, "Efficient optimization of common base domains for cross parameterization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 10, pp. 1678–1692, 2012.
- [20] M. Tarini, K. Hormann, P. Cignoni, and C. Montani, "Polycube-maps," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 853–860, 2004.
- [21] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or, "Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 126:1–126:10, 2011.
- [22] Y. Wang, S. Asafi, O. van Kaick, H. Zhang, D. Cohen-Or, and B. Chen, "Active co-analysis of a set of shapes," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 165:1–165:10, 2012.
- [23] Q.-X. Huang, G.-X. Zhang, L. Gao, S.-M. Hu, A. Butscher, and L. Guibas, "An optimization approach for extracting and encoding consistent maps in a shape collection," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 167:1–167:11, 2012.
- [24] D. Cohen-Or, A. Solomovic, and D. Levin, "Three-dimensional distance field metamorphosis," *ACM Trans. Graph.*, vol. 17, no. 2, pp. 116–141, 1998.
- [25] G. Turk and J. F. O'Brien, "Shape transformation using variational implicit functions," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99, 1999, pp. 335–342.
- [26] G. Turk and J. F. O'Brien, "Modelling with implicit surfaces that interpolate," *ACM Trans. Graph.*, vol. 21, no. 4, pp. 855–873, 2002.
- [27] J. Lin, X. Jin, C. C. L. Wang, and K.-C. Hui, "Mesh composition on models with arbitrary boundary topology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 653–665, 2008.
- [28] L. StăNculescu, R. Chaine, M.-P. Cani, and K. Singh, "Sculpting multi-dimensional nested structures," *Comput. Graph.*, vol. 37, no. 6, pp. 753–763, 2013.
- [29] F. Berthouzoz, A. Garg, D. M. Kaufman, E. Grinspun, and M. Agrawala, "Parsing sewing patterns into 3d garments," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 85:1–85:12, 2013.
- [30] E. Turquin, J. Wither, L. Boissieux, M.-P. Cani, and J. F. Hughes, "A sketch-based interface for clothing virtual characters," *IEEE Comput. Graph. Appl.*, vol. 27, no. 1, pp. 72–81, 2007.
- [31] C. C. L. Wang, Y. Zhang, and H. Sheung, "From designing products to fabricating them from planar materials," *IEEE Comput. Graph. Appl.*, vol. 30, no. 6, pp. 74–85, 2010.
- [32] C. Yuksel, J. M. Kaldor, D. L. James, and S. Marschner, "Stitch meshes for modeling knitted clothing with yarn-level detail," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 37:1–37:12, 2012.
- [33] M. Skouras, B. Thomaszewski, P. Kaufmann, A. Garg, B. Bickel, E. Grinspun, and M. Gross, "Designing inflatable structures," *ACM Trans. Graph.*, vol. 32, no. 4, 2014.
- [34] P. Guan, L. Reiss, D. A. Hirshberg, A. Weiss, and M. J. Black, "Drape: Dressing any person," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 35:1–35:10, 2012.
- [35] R. Brouet, A. Sheffer, L. Boissieux, and M.-P. Cani, "Design preserving garment transfer," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 36:1–36:11, 2012.
- [36] Y. Meng, C. C. L. Wang, and X. Jin, "Flexible shape control for automatic resizing of apparel products," *Comput. Aided Des.*, vol. 44, no. 1, pp. 68–76, 2012.
- [37] Y. Mori and T. Igarashi, "Plushie: An interactive design system for plush toys," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.
- [38] J. Mitani and H. Suzuki, "Making papercraft toys from meshes using strip-based approximate unfolding," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 259–263, Aug. 2004.
- [39] I. Shatz, A. Tal, and G. Leifman, "Paper craft models from meshes," *Vis. Comput.*, vol. 22, no. 9, pp. 825–834, 2006.
- [40] <http://www.wojumps.com/jq/3158700.aspx>.
- [41] M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe, "Design of tangent vector fields," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.
- [42] A. Loomis, *Figure drawing for all it's worth*. Viking Adult, 1943.
- [43] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe, "Fast exact and approximate geodesics on meshes," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 553–560, 2005.
- [44] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '97, 1997, pp. 209–216.
- [45] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: A sketching interface for 3d freeform design," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 409–416.
- [46] L. Prasad, "Morphological analysis of shapes," *CNLS Newsletter*, vol. 139, pp. 1–18, 1997.
- [47] A. Jacobson, I. Baran, J. Popović, and O. Sorkine, "Bounded biharmonic weights for real-time deformation," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 78:1–78:8, Jul. 2011.
- [48] S. Vogel, *Life's Devices: The Physical World of Animals and Plants*. Princeton University Press, 1988.
- [49] <http://en.wikipedia.org/wiki/Langer>



Tsz-Ho Kwok is currently a Postdoctoral Research Associate in the Epstein Department of Industrial and Systems Engineering from The University of Southern California. He received his B.Eng. degree in the Department of Automation and Computer-Aided Engineering, and Ph.D. degree in the Department of Mechanical and Automation Engineering, from the Chinese University of Hong Kong. His research interests include CAD/CAM, geometric and solid modeling, human-centric product design, image processing, and 3D printing.



Yan-Qiu Zhang is a postgraduate student at Tsinghua University, Computer Science and Technology Department, Beijing, China. She graduated in Computer Science and Technology Department from Beijing University of Technology in 2008. Her research focuses on computer graphics and video summarization.



Charlie C.L. Wang is currently an Associate Professor at the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong. He gained his B.Eng. in Mechatronics Engineering from Huazhong University of Science and Technology, M.Phil. and Ph.D. in Mechanical Engineering from The Hong Kong University of Science and Technology. He is a Fellow of ASME. His research interests are geometric modeling, design and manufacturing, and

computational physics.



Computer Society.

Yong-Jin Liu received his B.Eng. degree from Tianjin University, China, in 1998, and his Ph.D degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2004. He is now an Associate Professor with the TNList, Department of Computer Science and Technology, Tsinghua University, China. His research interests include computational geometry, computer graphics and computer-aided design. He is a member of IEEE, a member of IEEE



Kai Tang is currently a faculty member in the Department of Mechanical and Aerospace Engineering at Hong Kong University of Science and Technology. Before joining HKUST in 2001, he had worked for more than 13 years in the CAD/CAM and IT industries. His research interests concentrate on designing efficient and practical algorithms for solving real-world computational, geometric, and numerical problems. Dr. Tang received Ph.D. in Computer Engineering from the University of Michigan in 1990, M.Sc. in Information and Control Engineering in 1986, also from the University of Michigan, and B.Eng. in Mechanical Engineering from Nanjing Institute of Technology in China in 1982.