Sketch-Based Aesthetic Product Form Exploration from Existing Images Using Piecewise Clothoid Curves

Günay Orbay, Mehmet Ersin Yümer, Levent Burak Kara*
Mechanical Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213
Email: *lkara@cmu.edu

Abstract

We present a new sketch-based product form exploration technique that works from images and sketches of existing products. At the heart of our approach, is a multi-stroke curve beautification method and a curve-based image deformation algorithm. The proposed approach converts groups of strokes into piecewise clothoid curves in order to produce visually pleasing shapes. The deformation diffusion algorithm then spatially distributes the user specified deformations through out the image to produce smooth transformations from the original image to the resulting image. We demonstrate the technique on a variety of images including photo-realistic images, real product images, and sketches.

Keywords: sketching; product design; aesthetics; curve design; image editing

1. Introduction

Image editing tools are becoming increasingly more ubiquitous with the wide availability of a variety of photo editing software [1]. However, such tools remain difficult to master due to the complex nature of their user interaction methods and the parameters that need to be controlled to achieve the desired effects. While many projects no doubt require advanced skills, experience, and appropriate task planning, even relatively simple modifications become difficult to implement with such tools. As a result, the manipulation and exploration of shape ideas using existing visual references remain a significant challenge in product design, forcing designers to generate a new sketch or a visual for each of their new ideas.

In this paper, we propose a new sketch-based method for the rapid exploration of new product forms from existing images and sketches. The proposed method enables the sketch input to control image deformations in which the visual attributes contained in the original image or sketch are preserved. By allowing image modifications to be made through simple sketched strokes, it reduces the effort required in manipulating high fidelity images and sketches. Moreover, it advances existing image editing methods which use fixed deformation masks, pick/drag type curve-based deformations, or area-based deformation lattices.

At the heart of our approach is an aesthetic curve construction method that works from multiple strokes, followed by a deformation field calculated by a diffusion algorithm. To achieve the desired visual properties, we use the previously identified forms of aesthetic curves, specifically piecewise *clothoid* curves. Figure 1 shows the usage of our system. Shape variations can be intuitively and fluidly explored on different kinds of images including photo-realistic images, real product images, and sketches. The user starts from an image or a sketch of an existing product in order to explore new designs. The user draws a set of strokes via a tablet interface to define a number of anchor curves, and source and target curve pairs which act as deformation handles for the editing. Each group of strokes that define an individual curve, is converted into a single curve with the help of a point ordering algorithm. The source and anchor curves are aligned with the underlying features using active contours, while the target curves are converted into piecewise clothoid curves. The input image is then deformed using a smooth deformation field that is calculated from the source and target curve pairs. As the target curves are designed to be visually pleasing, the deformed features in the final image also carry similar properties.

2. Contributions

The proposed work makes specific contributions in utilizing sketch-based interaction in rapid product form exploration, and in technical methods that enable this exploration. The first contribution of the proposed method is that it provides a means to utilize intuitive sketching input in exploring visually pleasing design forms from images and sketches of existing products. This, in return, allows the user to still benefit from the intuitive and natural way of externalizing shape ideas using the sketch input. Moreover, the user focuses on the resulting product form directly, instead of obtrusively paying attention to the visual quality and accuracy of each drawn stroke.

The second contribution is that the proposed method enables the conversion of overlapping strokes to be seamlessly converted into piecewise clothoid curves. This allows the user

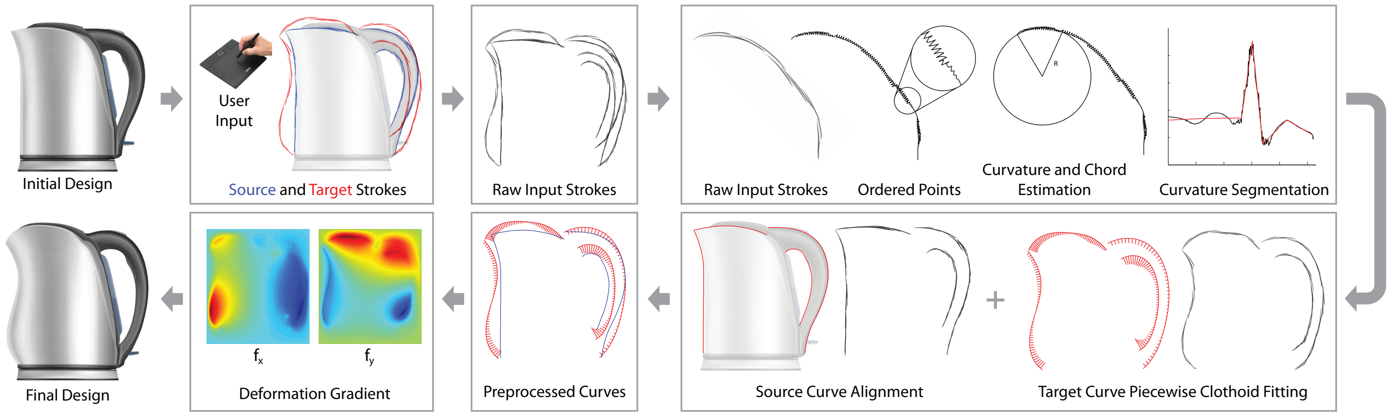

Figure 1: The overview of the proposed method. The user inputs an image of an existing product in order to explore new designs. The user then inputs a set of strokes via a tablet interface to define a number of source curves and corresponding target curves which act as image controllers. These input strokes are preprocessed into desirable computational forms through a series of operations. The input image is then deformed into the final design using a smooth deformation field that is calculated form the source and target curve pairs.

to work on the design using the natural sketch input without putting emphasis on the accuracy of the strokes. The last contribution is that the proposed method produces images that contain visually pleasing features and shapes as it grabs the initial features in the image and deforms them into piecewise clothoid curves with a smooth deformation field.

3. Related Work

3.1. Image Editing

The main focus in image editing have been in cut-and-paste type operations and image enhancing. Elder *et al.* [2] presented a contour based image editing technique where modifications are applied directly in the image contour domain. These modifications are then transferred to the final image using edge based image reconstruction [3]. This approach enables robust cut-and-paste type operations. Saund *et al.* [4] have introduced a perceptually-supported editing tool where they combine a variety of previously studied image editing methods [5, 6] deployed in an interactive environment. In an earlier work [6], the authors studied decomposing simple images into geometric primitive based shapes for enabling simple shape based editing. In 2002, Seitz *et al.* [7] introduced a set of image editing techniques which enable a sequence of images to be modified through operations applied to one of the images in the sequence. Their method makes use of the plenoptic function through which the modifications are propagated in a physically-consistent way. Perez *et al.* [8] presented a set of image editing operations based on second order discrete Poisson equations in image domain. Their approach enables seamless region mapping between images and modification of original illumination, color and texture in arbitrary regions of a given image, our method focuses on nonlinear warping of a single image through user defined strokes.

3.2. Curve-Based Image Deformation

Closely related to our work, Lee *et al.* [9] proposed an image morphing method that provides an easy-to-use means to specify, deform and morph features. They similarly utilize active contours to accurately grab features and a multilevel free-form deformation algorithm to deform those features. Likewise, Fang *et al.* [10] developed a curve-based image deformation tool mainly focused on texture regeneration on distorted regions. Shungang *et al.* [11] used cubic splines and Moving Least Squares for image deformation using curves as handles. In this work, however, we attempt to extend curve-based image deformation methods by combining with sketch-based curve design in order to utilize in rapid product form design.

3.3. Piecewise Clothoid Fitting

Different approaches have been proposed for fitting clothoid curves into a set of organized and unorganized points [12, 13, 14, 15, 16]. Pal and Nutbourne [12], Schechter [13] a year later, proposed methods that fit piecewise clothoids given two end points with specified positions, tangents and curvatures. Stoer [14] developed a numerical algorithm to fit single and piecewise clothoids to points. Meek and Walton [15] described the algorithms that match curvatures and tangents of multiple clothoids at their attached ends. Davis [16] described a rapidly convergent total least-squares algorithm. For a broader collection of clothoid fitting methods [17] can be referred.

Related to our sketch-based piecewise clothoid fitting approach is the work by Harada and Yoshimoto [18]. They applied b-spline fitting methods on point sets and then converted them into piecewise monotonic curvature curves. McCrae and Singh [19], developed a sketching interface that converts individual strokes into piecewise clothoids for road design. They based their approach on a curvature analysis followed by a segmentation algorithm. Similarly, Baran *et al.* [20] introduced a graph based method that identifies an optimum path of local fits that results in a piecewise curve composed of lines, arcs and clothoids. Although these approaches are successful in processing individual strokes, they do not readily support curve fits from a group of overlapping strokes.

3.4. *Differences from Existing Work*

Our approach differs from the above works in a number of ways. Our main goal is to enable rapid sketch-based exploration of new product forms from existing product images and sketches. Specifically, we attempt to utilize conventional sketching in design of fair curves, and use these curves to control image deformations that result in visually pleasing shapes. As a result, our approach attempts to enable rapid shape exploration using the convenient sketch input from largely available images and sketches of existing products.

4. Overview and User Interaction

Figure 1 illustrates the details of our system. The input to our system is an image and a set of user-drawn strokes which define deformation handle curves. The users may draw strokes to define curves in three categories:

Anchor Curves: (Orange color) These curves indicate regions of the image that the yser wants to preserve unchanged. Any deformation ceases at these curves.

Source Curves: (Blue color) These curves demarcate features in the image that will undergo user-specified deformations. Input source curves are first modified under local image gradients that precisely align them with the salient image features.

Target Curves: (Green color) For each curve stroke, a target curve indicates the new desired shape. Each target curves is automatically beautified into a piecewise clothoid curve using curvature estimation, segmentation, and curve fitting to the original strokes.

Once processed, stroke groups in each category are internally represented as curves of the same category. Given the anchor, source, and target curves, a smooth vector deformation field is calculated on the image. The computed deformation field is then applied to the image, resulting in the modified image. The key property of the deformation is that the image attains the precise shape dictated by the target curves, while the remainder of the image undergoes a locally smooth deformation. The users repeat this process as many times as necessary, which results in the final form of the design.

5. Preprocessing the Input

The input to our system is a set of user-drawn strokes that define curves in three categories: (1) source curves, (2) target curves, (3) anchor curves. During sketching, the users typically draw groups of overlapping strokes each of which is intended to represent a single curve (Fig. 2.a). From this input, our approach seeks to produce curves that satisfy the following three requirements. First, each stroke group must be converted into an individual curve that closely approximates the strokes in the group, regardless of its category. Second, the calculated source curves must be accurately aligned with the salient feature edges in the image. This guarantees that the features in the image will properly deform into the shapes of the target curves. Third, the calculated target curves must be beautified into visually pleasing curves so as to eliminate the artifacts that may arise from a literal interpretation of the raw strokes.

5.1. *Multiple Stroke Analysis*

In the first step of preprocessing, our approach first analyzes each stroke group and converts it into a single curve. To be able to utilize existing curve fitting methods, it is necessary to determine an estimated direction and length of the curve. Note that incoming strokes can be drawn in arbitrary directions and order. Hence, a global analysis of the sampled points that comprise the constituent strokes is necessary to identify a natural spatial order for the sampled points, and hence the strokes. Both cases, however, require that the sampled points comprising the input strokes are pre-processed to identify a natural spatial ordering and a chord length parameterization of the points. As demonstrated later, this ordering facilitates an energy-based feature edge identification as well as curve synthesis from a curvature profile. Our approach uses a spectral point ordering method and a moving circle analysis to support curve fitting calculations on both the source and the target curves. This analysis produces an accurate estimate of the chord length parametrization (thus an approximate curve length) as well as a noise free estimate of the curvature along the curve.

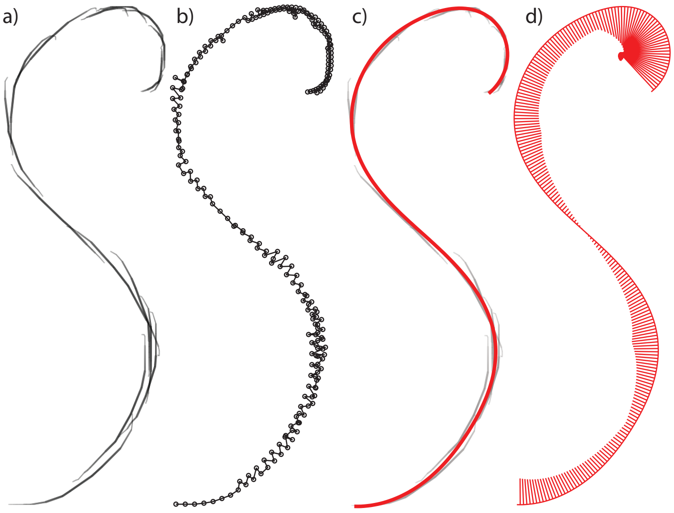

Figure 2: (a) A curve consisting of multiple strokes are analyzed to identify a (b) natural ordering of the points that make up the original strokes. (c) The identified ordering facilitates fitting of a piecewise clothoid curve that closely approximates the original strokes. (d) The curvature profile of the final curve is locally linear except at the transitions between the neighboring clothoid segments.

We start by collecting the points sampled along each stroke into a point set. We utilize the spectral point ordering method proposed by Orbay and Kara [21] to calculate an initial ordering of the points. Once this ordering is calculated we next estimate the chord length parametrization and the curvature profile suggested by the ordered points. To do so, we use a moving circle approach which simultaneously calculates local estimates of the chord length parameter and the curvature. As shown in Fig. 3, a least-squares circle is fit to the points located inside a sliding window. Using the vectors formed by the circle center and each of the sampled points, the points are ordered within the window and assigned a chord-length parameter. For each sliding window, we use the weights of a Hamming window as the confidence values of the chord length, and the curvature (*i.e.* $1/R$). The weights of the Hamming window is calculated as follows:

$$w(s) = 0.54 - 0.46 \cos\left(2\pi \frac{sN+1}{N-1}\right) \tag{1}$$

where $s = [0, 1]$ is the normalized chord length within the window, and N is the number of points in the window. The closer the points are to the middle of the sliding window the higher confidence value they attain. The final estimates are calculated over the aggregate local estimates and their associated aggregate confidence values. An example chord length and curvature estimate for the stroke group shown in Fig. 2.a is shown in Fig. 4.

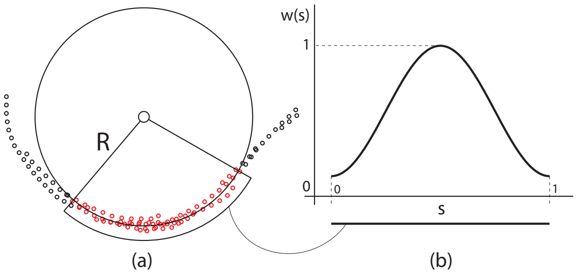

Figure 3: The curvature and chord length parameters are estimated simultaneously using a sliding window over the initially ordered points using a spectral analysis. (a) A circle is fit within each window and the radius of the circle and the chord length on the circle circumference is computed. (b) The Hamming window function used in the calculation of the point weights.

5.2. *Source Curves: Alignment on to Feature Edges*

In this step, we extract and align the source curves on to the underlying features of the image. Before we begin this process, we first define a polyline using the chord length estimation that we calculated in the previous step. Before we align the polyline on to the feature, we smooth the curve to eliminate the jaggedness of the initial configuration (Fig. 2.b).

We use active contours [22] to gradually pull the source curves on to the underlying features of the image. For this, an edge map is extracted from the image using Canny Edge Detection [23], and the points comprising the polyline are pulled toward the identified edges. The process is initialized by anchoring the end points of the source curve to the nearest stable points on the edge map, and using the regular active contours formulation to achieve a smooth curve that is aligned with the underlying feature.

5.3. *Target Curves: Piecewise Clothoid Fitting*

To achieve the desired visual properties on the deformed images, we make use of previously identified aesthetic curve forms to beautify the target curves. Recent studies [24, 25, 26] have demonstrated that curves with piecewise linear logarithmic curvature histograms (LCH) exhibit visual appeal. In our approach, we use one of many possible such curves, namely the piecewise *clothoid* curves with piecewise linear curvature profiles. In this step, we beautify the target strokes into piecewise clothoid curves through a series of operations. We first begin by identifying a segmentation of the curve aided by the previously calculated curvature and chord length estimation. We then use a numerical fitting algorithm that minimizes the spatial error between the original points and the piecewise clothoid curve by controlling the curvature values at the segmentation points.

5.3.1. *Curvature Segmentation*

Clothoid curves, as their curvature profiles are restricted to be linearly changing, can represent only a limited range of shapes. They have at most one inflection point and a constant rate of change of the curvature with respect to the chord length. As a consequence, for a proper representation of a curve exhibiting rather complicated curvature profiles with clothoids, it is of utmost importance to decide where to break the curve and what number of segments to use. This decision directly dictates the trade off between the overall smoothness of the curve and the approximation accuracy of the points. On one extreme, using only one clothoid curve is globally smooth yet not able to represent complicated curves. On the other extreme, having a large number of clothoid segments results in rapidly changing trend in the curvature profiles thus leading to visually less pleasing results. To be able to control this trade off with a user defined parameter, we adopted the curvature segmentation method proposed by McCrae and Singh [19].

A matrix of costs of the optimal segmentations between any given two points is calculated in a bottom-up fashion using the following cost function as defined in the original work [19]:

$$M(a, b) = \min_{a<k<b} \{M(a, k) + M(k, b), E_{fit}(a, b) + E_{cost}\} \tag{2}$$

where $M(a, b)$ represents the minimal fit error of the associated configuration of line segments between points a and b. The fitting error is simply as the sum of absolute errors as

$$E_{fit}(a, b) = \sum_{i=a}^{b} ||\kappa_{fit}(u_i) - \kappa_i|| \tag{3}$$

where u_i, κ_i and $\kappa_{fit}(u_i)$ are the estimated chord length, estimated curvature, and the fit curvature at point i, respectively. E_{cost} is the cost per line segment in the curvature space and indirectly controls the total number of segments in the minimal fit case. This parameter is of paramount importance since the number of segments dictate the quality of the approximation to the input strokes. We will have detailed discussions on how this parameter controls the compromise between the fidelity of the approximation and the resulting level of global smoothness.

5.3.2. *Piecewise Clothoid Fitting*

The result of the previous curvatures segmentation step is a list of curvatures (κ_i) and associated chord length values (u_i) which represent the breaking points of a linearly piecewise curvature profile. In this section we describe and derive a numerical fitting method that works specifically from a previously cal-

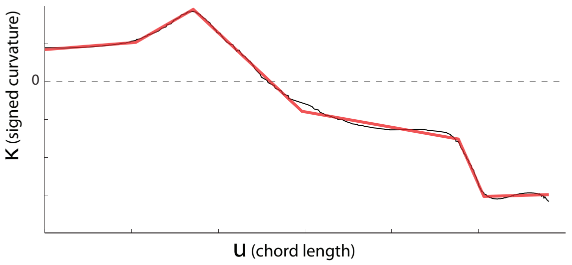

Figure 4: The final segmentation in the curvature space of strokes given in Fig. 2.a. The resulting number of clothoid segments is controlled by a single parameter (E_{cost} in Eqn. 2).

culated segmentation in the curvature space[1]. Mathematically, the curvature profile can be defined using piecewise linear interpolation functions as:

$$\kappa(u_i) = \sum_{n=1}^{N_\kappa} \kappa_n \Phi_n(u_i) \tag{4}$$

where

$$\Phi_n(u_i) = \begin{cases} \dfrac{u - u_{i-1}}{u_i - u_{i-1}} & \text{if} \quad u \in [u_{i-1}, u_i] \\ \dfrac{u_{i-1} - u}{u_{i+1} - u_i} & \text{if} \quad u \in [u_i, u_{i+1}] \\ 0 & \text{elsewhere} \end{cases} \tag{5}$$

Using this curvature function, we first write the angle in terms of the chord length as a summation:

$$\theta(u_i) = \sum_{m=1}^{i} \kappa(u_m)\Delta u_m \tag{6}$$

where $\theta(0) = 0$, $\Delta u_m = u_m - u_{m-1}$ and $\Delta u_1 = 0$. Similarly, we write the spatial coordinates of the associated points on the curve as a summation:

$$x(u_i) = \sum_{m=1}^{i} \cos(\theta(u_m))\Delta u_m \tag{7}$$

$$y(u_i) = \sum_{m=1}^{i} \sin(\theta(u_m))\Delta u_m \tag{8}$$

where $x(u_i) = y(u_i) = 0$.

Given a set of points (x_p, y_p) and a piecewise clothoid curve, we define a spatial error function to be minimized. For a reliable error definition, we first align the original points and the piecewise clothoid curve. To do so, we translate both the curve and the points such that their centroids are located at the origin. We then calculate a transformation that aligns the original points on to the associated points on the curve as dictated by the chord length parameter.

$$[x_c \;\; y_c] = [T][x_p \;\; y_p] \tag{9}$$

This transformation includes not only the necessary amount of rotation for alignment, but also the scaling and shearing effects. For the alignment, however, we only use the rotation part using polar decomposition [27] as:

$$[T] = [R][S] \tag{10}$$

We then transform the original points using the decomposed rotation part of the initially calculated transformation:

$$[\bar{x}_p \;\; \bar{y}_p] = [T][x_p \;\; y_p] \tag{11}$$

Here, we define the error as:

$$E = \sum_{i=1}^{N} (\bar{x}_p^i - x_c^i)^2 + (\bar{y}_p^i - y_c^i)^2 \tag{12}$$

Minimizing E can be achieved by minimizing a residual vector defined as:

$$\{R\} = \left\{ \begin{array}{c} \bar{x}_p - x_c \\ \bar{y}_p - y_c \end{array} \right\} \tag{13}$$

Since the spatial coordinates of the clothoid curve is a nonlinear function of the curvature, minimizing R requires an iterative solution. For this, we use the following Newton-Raphson iteration:

$$\{\kappa_{i+1}\} = \{\kappa_i\} - \left[\frac{\partial\{R\}}{\partial\{\kappa\}}\right]_i^{-1} \{R_i\} \tag{14}$$

The Jacobian $\left[\frac{\partial\{R\}}{\partial\{\kappa\}}\right]$ can be derived using Equations 7 and 8 as:

$$\frac{\partial\{R\}}{\partial\{\kappa\}} = -\left[\begin{array}{c} J^x \\ J^y \end{array} \right] \tag{15}$$

where

$$J^x_{i,j} = \frac{\partial x_i}{\partial \kappa_j} = -\sum_{m=1}^{i} \sin(\theta(u_m)) \frac{\theta(u_m)}{\partial \kappa_j} \tag{16}$$

$$J^y_{i,j} = \frac{\partial y_i}{\partial \kappa_j} = \sum_{m=1}^{i} \cos(\theta(u_m)) \frac{\theta(u_m)}{\partial \kappa_j} \tag{17}$$

and from Equation 4 and 6

$$\frac{\theta(u_m)}{\partial \kappa_j} = \sum_{k=1}^{m} \phi_j(u_k)\Delta u_k \tag{18}$$

We use the curvature values calculated at the curvature segmentation step as the initial values for the minimization. At each iteration of Eqn. 14, we calculate a different transformation as defined in Eqn. 11. Since the coefficients of the curvature values are functions of the estimated chord length parameter, the Jacobians can be calculated with matrix operations.

The described minimization yields the piecewise clothoid curve that has the same segmentation as calculated in the previous step and it minimizes the spatial error between the original points and the curve. However, as discussed earlier, the amount of the minimized error is directly dictated by the curvature segmentation. Figure 5 demonstrates the trade off between accuracy (*i.e.* the spatial error) and overall smoothness of the

[1] Please see the Appendix for more details on the clothoids.

curve. For the set of strokes shown (Fig. 5.a), using a high number of segments can cause undesirable fluctuations in the curvature profile (Fig. 5.b), whereas using very few segments can severely affect the approximation accuracy (Fig. 5.f,g). We leave the control over this trade off to the user and let them manually control the cost parameter that dictates the resulting curvature segmentation.

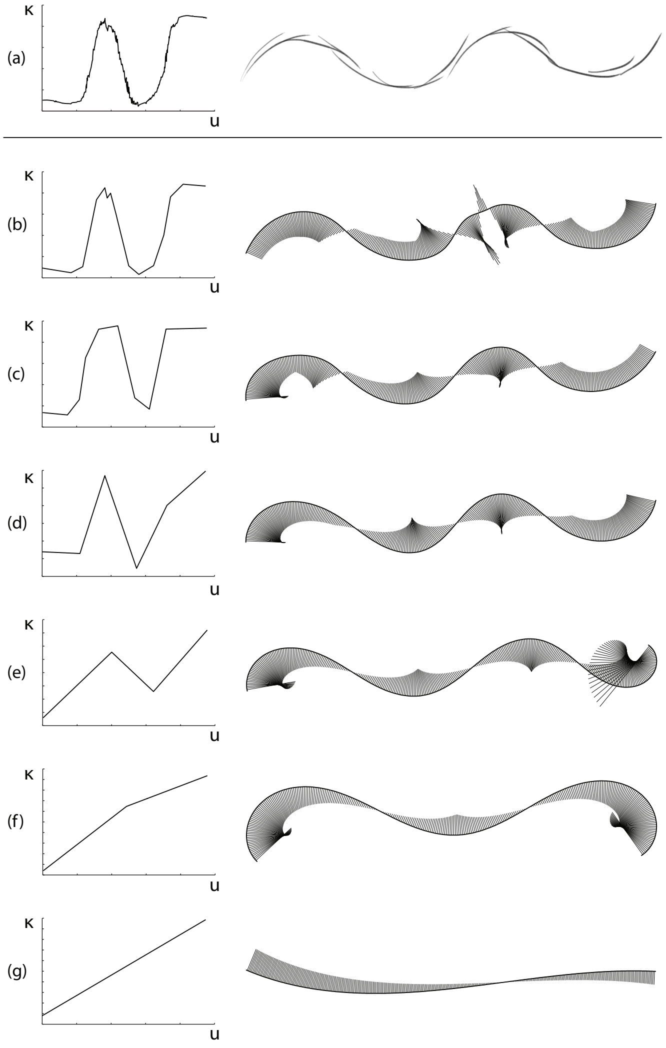

Figure 5: (a) The curvature space (*i.e.* the estimated chord length vs. curvature) of the initial point set is used to compare the effect of the number of piecewise clothoid segments. This number is directly controlled by the cost parameter, $E_{c}ost$. As the cost parameter increases from (b) to (g), the number of segments thus the approximation accuracy increases. However, the visual appeal of the fits with higher number of segments becomes debatable as the fit approaches to the strokes which are originally overtraced and noisy. (g) On the other hand, the low number of segments result in weak approximations to the strokes. The compromise between the approximation accuracy and the visual appeal is to be decided by the user.

6. Curve-Based Image Deformation

The input to the image deformation step is a set of preprocessed source and target curves, and the set of anchor curves. This step calculates a smooth spatial deformation that transforms each source curve into the corresponding target curve, while preserving the anchor curves. For this purpose, we use an isotropic vector diffusion method to spatially diffuse the deformation vectors between the source and target curve pairs.

Although there are arbitrarily many vector fields that can produce the desired deformation, in this work we use a formulation similar to the gradient vector flow field (GVF) method [28]. This approach suitably diffuses the initial deformation vectors computed between each source-target curve pair, thus producing deformations that vary smoothly through the image space.

The GVF is defined as the continuous 2D vector field $\vec{V}(x,y) = [u(x,y), v(x,y)]$ that minimizes the following energy functional [28]:

$$\epsilon = \iint (\underbrace{\mu(u_x^2 + u_y^2 + v_x^2 + v_y^2)}_{\text{Smoothness}} + \underbrace{\|\triangledown f\|^2 \|\vec{V} - \triangledown f\|^2}_{\text{Pointing objects}}) dxdy \tag{19}$$

where $f(x,y)$ is a 2D scalar field (*e.g.* black and white scanned medical images in the original work), $\triangledown$ is the gradient operator in Cartesian coordinates, $\vec{u}(x,y)$ and $\vec{v}(x,y)$ are the x and y components of the vector field at point (x,y) while the subscripts denote partial derivatives. μ is the coefficient which controls the amount of diffusion. In this energy functional, the first term forces the local smoothness of the calculated vector field while the second term forces $\vec{V}(x,y)$ to converge to the original deformation vector in regions of high deformation. The vector field $\vec{V}(x,y)$ minimizing this energy functional can be calculated using calculus of variation by solving the following two decoupled Euler equations:

$$\begin{aligned} \mu\triangledown^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \\ \mu\triangledown^2 v - (v - f_y)(f_x^2 + f_y^2) = 0 \end{aligned} \tag{20}$$

where the Laplacian operator is defined as:

$$\triangledown^2 u(x,y) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{21}$$

The two discretized equations can be solved independently to produce the vector field components u and v separately. In this application, the discretization of the domain naturally comes from the pixelated image. In our implementation, we use the deformation vectors drawn between source and target curves to instantiate the f_x and f_y values in Eqn. 20. An iterative multigrid solver that enables an efficient computation of the solution to the differential equations yields the resulting vector field. We note, however, that any conventional differential equation solver can be similarly adopted.

The boundary conditions to the above set of equations arise from the anchor curves (zero deformation), and the deformation vectors from each source curve to the corresponding target curves. These vectors are preserved throughout the computation. The remainder of the vector field is computed using the above two equations.

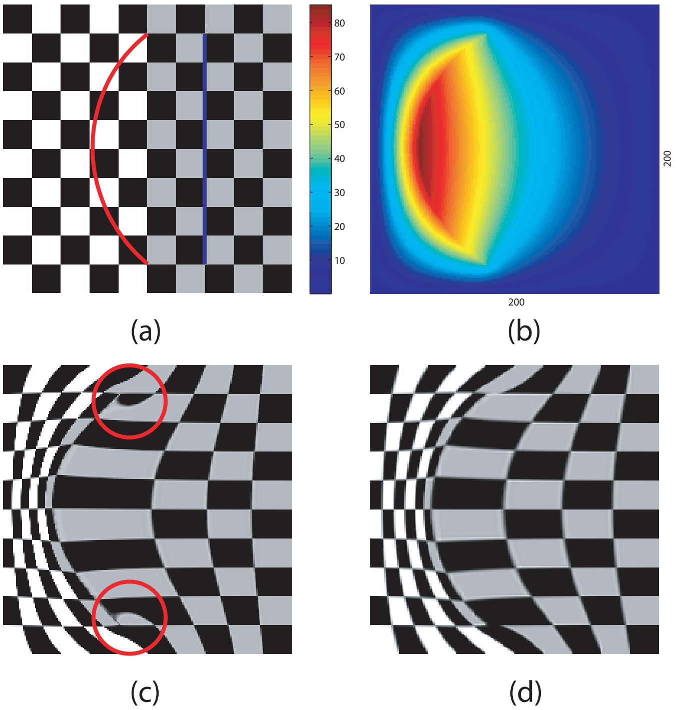

Figure 6: Calculation of a spatial transformation field from the source and target curves. (a) The displacement vectors from the source curves (blue) to the target curves (red) are diffused to achieve a smooth deformation field. (b) The calculated spatial deformation field of displacements in the x direction. (c) The final image deformed by the calculated deformation field.

Prior to initializing the boundary conditions, we resample the target and source curves to have the same number of points along their paths. The sampling density is chosen such that the distance between adjacent sample points is always less than the pixel length. This ensures all pixels along the curves to attain a unique deformation vector. Moreover, a simple direction check is performed to map each end of the source curve to the intended end of the target curve, thereby eliminating sensitivity to drawing directions.

A problem arises if the computed deformation field is applied directly to the original image, as the deformation vectors do not guarantee a complete mapping from the original image to the yet-to-be computed deformed image (Fig. 6.c). To overcome this difficulty, we utilize deformation vectors that originate from the target curves and point toward the source curves. The subsequent diffusion is then applied in this new configuration. This allows, for every pixel in the deformed image to point to a real-valued location in the original image (through the deformation vector) and attain a value as an interpolation from the original image. To avoid aliasing effects, we use a bicubic spline interpolation. Figure 6 shows intermediate steps of deformation for two examples.

Another important property of the resulting deformation field is that, its gradient is discontinuous around the target curves. Figure 8 shows a 3D plot of the deformation field calculated for the example in Fig. 6 to illustrate this property. This rapidly changing gradient necessitates that the source curves to be accurately aligned with the features on the image to avoid undesirable smudging around the features.

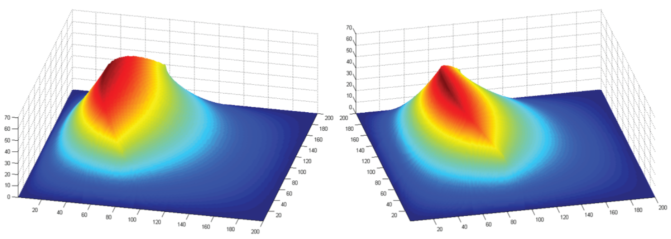

Figure 8: The deformation field is sharp around the curves with large gradients on the displacement values. In order to preserve the feature edges in the input image, it is crucial to have the source curves to be closely aligned with the underlying edges.

Figures 6 and 10 assess the nature of the deformations over circular and rectangular patterns. As shown, when circular patterns are forced into rectangular shapes, the internal and external regions are smoothly deformed yet they cannot preserve their perfect circular structures. Similarly, when rectangular patterns are forced into circular shapes, although the deformed features shape into the target curves, the neighboring rectangles neither form circles nor stay as rectangles. To selectively preserve such patterns, the users can use anchor curves to limit the effective area of the deformation as shown in Fig. 9.

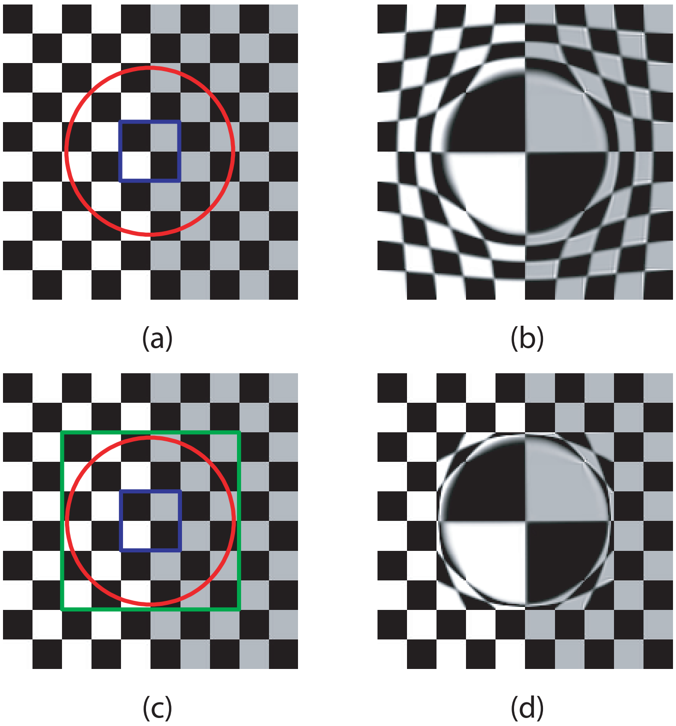

Figure 9: The anchor curves serve as limiters on propagation of the deformation field. The user controls the region that the deformation can affect by trapping it region with anchor curves.

Figure 10 further illustrates the effects of the deformation on the final curvature profiles. The checkers contained in a square are deformed such that the top edge is converted into a single clothoid. As shown, the horizontal lines which were previously

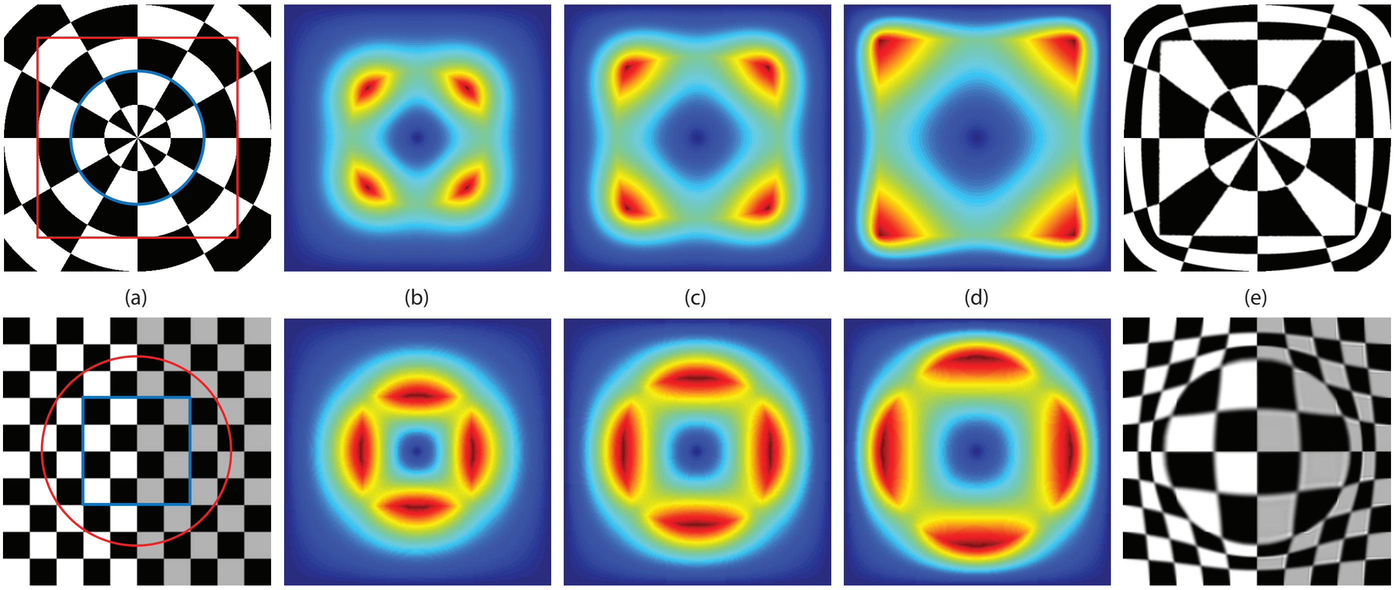

Figure 7: Checker board patters are deformed from rectilinear to circular and from circular to rectilinear patterns. (a) The original patterns undergo (b-d) multiple steps of deformations to yield the (e) final patterns. Note that the internal and external patterns show smooth shapes as the deformation field itself is smooth.

straight, attain similar curvature profiles as the final shape of the top edge. However, the curvatures deviate form a clothoid profile in the vicinity of the vertical edges at the left and right.

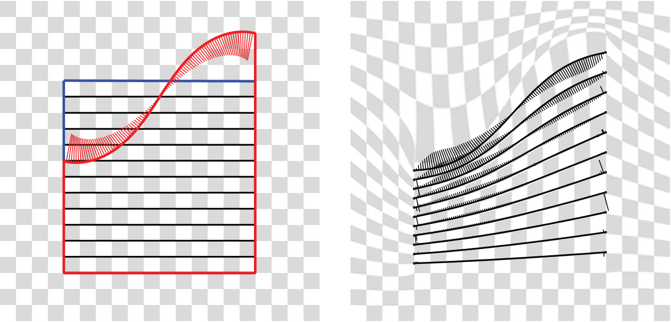

Figure 10: The effect of the deformation on the curvature profiles. (a) The originally flat lines are deformed into (b) still smooth but not clothoid curves.

7. Implementation and Examples

The proposed method is implemented in MATLAB. The user sketches the input curves on the original image. We are using a multigrid solver for the deformation field calculation. For a typical image of size 1024 × 768, the computation of the deformation field and its subsequent application to produce the deformed image takes on the order of 10 seconds on a dual core 2.5 GHz laptop with 3 GB of memory. However, especially when deformations are large, it might be desirable to apply the deformation in successive smaller steps, which will proportionally increase the processing time. In all of our examples we applied the deformations in 8 steps.

In Figure 11, intended modifications are sketched in the form of anchor, source and target curves. The resulting deformation is shown in Figure 11(c). Note that the deformations cease at the regions marked by the anchor curves. However, there are discrepancies between the target strokes and their beautified versions. These large discrepancies are the result of fitting single clothoids on the target curves, i.e. each target curve is a single clothoid. Figures 12 and Fig. 14 present similar modifications for a mug and a sofa, respectively. As described, all target curves are beautified into piecewise clothoid curves. We adjusted the number of segments manually to have a desired level of curve complexity that leads to visually pleasing forms. For instance, in Figure 14, we purposefully fit single clothoids to the target strokes to get the least amount of curvature fluctuations (Figure 14(c)).

Our method is also applicable to hand-drawn sketches. Figure 13 shows edits on a manga character. Figure A.15 illustrates a series of progressive edits on a car concept sketch. The sketch is first modified through a set of user drawn curves that define the modifications and constraints on the image (Figure A.15(b)). The result of these modifications (Figure A.15(c)) have gone through a second set of modifications (Figure A.15(d)), resulting in the final image (Figure A.15(e)).

Although our system handles a wide variety of images and sketches, exceptions exist. The approach is not designed to properly handle images with apparent texture elements, thus may cause texture distortions at undesirable magnitudes. A remedy would be combining it with texture analysis and synthesis methods as it has been proposed in [10]. However, for shape exploration in product design this shortcoming has limited effect.

Likewise, for local modifications the user is required to limit the deformation using anchor curves. This process can be fa-

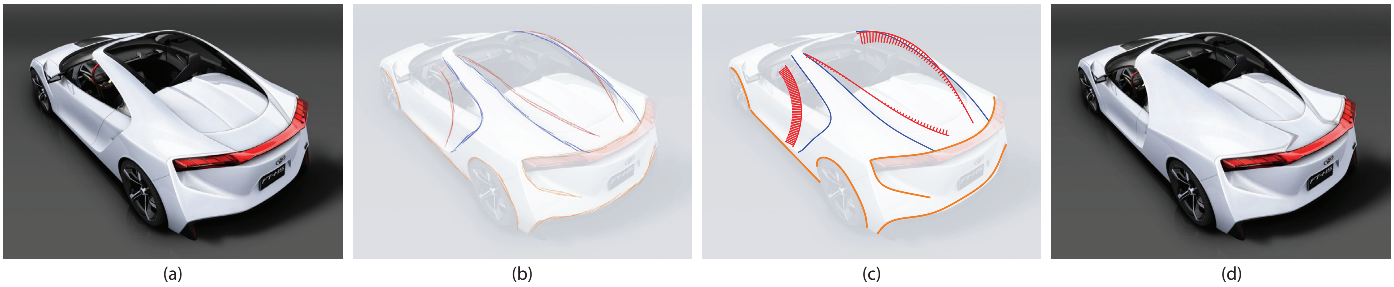

Figure 11: (a) A car design. (b) The source, target and anchor strokes drawn by the user are given by blue, red, and orange colors, respectively. (c) The source curves are aligned with features, and their corresponding target curves are beautified into clothoid curves. These curves define the intended modifications resulting in (d) the final design.

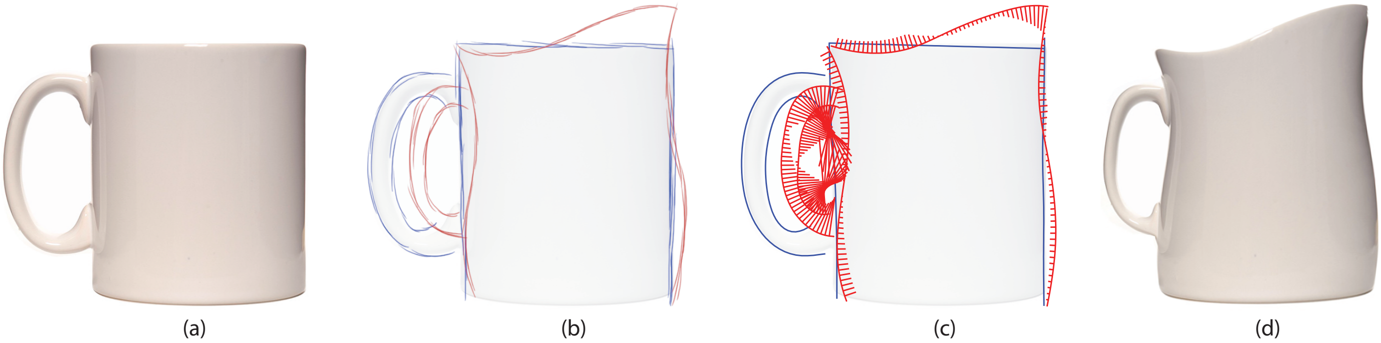

Figure 12: (a) A typical mug design. (b) The source, target and anchor strokes drawn by the user are given by blue, red, and orange colors, respectively. (c) The source curves are aligned with features, and their corresponding target curves are beautified into clothoid curves. These curves define the intended modifications resulting in (d) the final design.

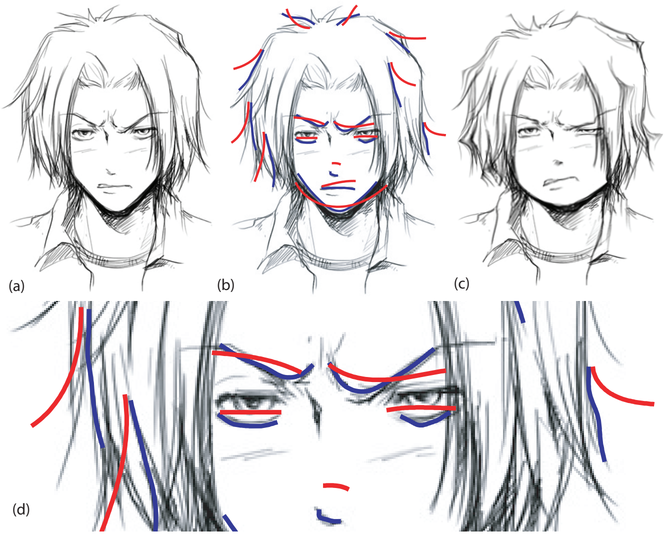

Figure 13: (a) The sketch of a manga character. (b) The set of curves that define the intended modifications. (c) The resulting modification. (d) The close up view of the source and target curve pairs. The colors blue, and red denote, the source curves, and the beautified target curves, respectively.

cilitated by a prior segmentation of the image into patches (*e.g.* the mug and its handle in Fig. 12). The patch under deformation would then be limited within itself not affecting the neighboring patches. Another extension would be an automated process of separating the actual object from the background for a more effective handling of the objects shape.

8. Conclusion

We present a sketch-based image deformation technique for exploration of visually pleasing product forms from existing images and sketches. Our method achieves the aesthetic qualities through piecewise clothoid fitting from multiple strokes, while deforming the image accordingly by a smooth deformation field. Our current studies indicate that the approach works effectively for a variety of different images and sketches.

Acknowledgements

All design images and sketches used in the paper are courtesy of their respective owners.

Appendix A. Clothoids and the Euler Spiral

An Euler spiral is defined as a curve whose signed magnitude of curvature changes linearly with its chord length. The typical Euler spider passes through the origin with zero curvature and approaches to two unique locations on the 2D plane. The curve can be defined as a differential equation such as:

$$\frac{d\theta}{ds} \propto s \tag{A.1}$$

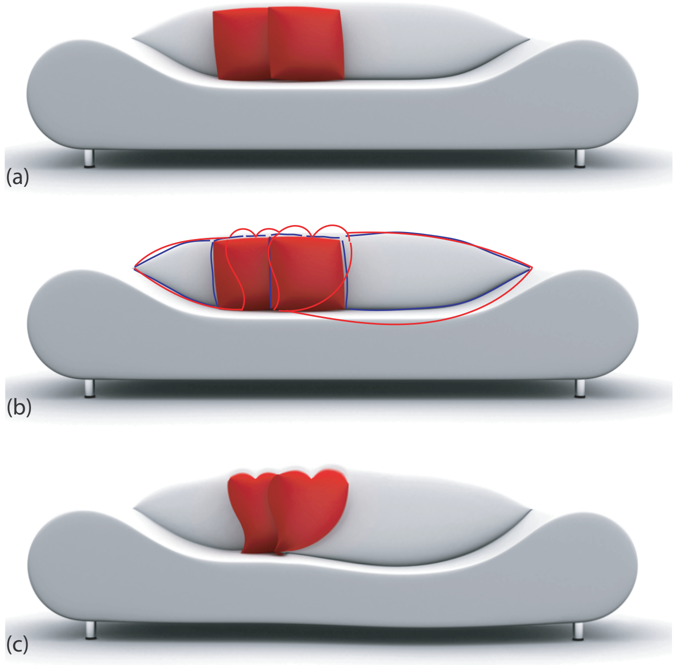

Figure 14: (a) A sofa design. (b) The source, target and anchor strokes drawn by the user are given by blue, red, and orange colors, respectively. The source curves are aligned with features, and their corresponding target curves are beautified into clothoid curves. (c) These curves define the intended modifications resulting in the final design.

where θ is the angle of the tangent of the curve, and s is the chord length. With different variables, the same equation can be written as:

$$\frac{d^2\vec{C}(s)}{ds^2} = \frac{d\vec{C}(s)}{ds}\left|\frac{d\vec{C}(s)}{ds}\right|^{-1} s \tag{A.2}$$

where $\vec{C}(s)$ is the position vector of the point on curve at chord length s. Each and every segment of the Euler spiral is defined to be a clothoid curve, sharing exactly the same properties.

Through integration, it can be shown that the position of any point on the curve does not have a closed solution. The coordinates can be represented as Fresnel integrals, or as a numerical approximation, it can be represented with power series expansions or numerical integration methods. In our formulations, we use a numerical integration scheme which calculates tangents and positions subsequently from finite differences in the chord length.

Compared to other commonly used parametric curve representations, the clothoid curves exhibit various differences. First of all, unlike polynomial function curves such as splines, clothoids cannot be controlled locally through a finite set of control points. Local modifications on the clothoids requires recalculating the governing equations as to produce local changes. This requirement alongside with the linearly changing curvature results in a constrained set of shapes that can be achieved. On the other hand, parametric curves typically require high number of control points to accurately produce linearly changing curvature profiles.

References

[1] Adobe Systems Incorporated, Photoshop, 2011. http://www.adobe.com/products/photoshop.html.

[2] J. Elder, R. Goldberg, Image editing in the contour domain, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 291–296.

[3] J. Elder, S. Zucker, Space scale localization, blur, and contour-based image coding, in: IEEE CVPR, pp. 27–34.

[4] E. Saund, D. Fleet, D. Larner, J. Mahoney, Perceptually-supported image editing of text and graphics, ACM Transactions on Graphics 23 (2004) 728.

[5] S. Bagley, G. Kopec, Editing Images of text, Communications of ACM 37 (2001) 63–72.

[6] E. Saund, T. P. Moran, A perceptually-supported sketch editor, Proceedings of the 7th annual ACM symposium on User interface software and technology - UIST '94 (1994) 175–184.

[7] S. Seitz, K. Kutulakos, Plenoptic image editing, in: Sixth International Conference on Computer Vision IEEE, Narosa Publishing House, 2002, pp. 17–24.

[8] P. Pérez, M. Gangnet, A. Blake, Poisson image editing, ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03 (2003) 313.

[9] S. Lee, K. Chwa, S. Shin, Image metamorphosis using snakes and free-form deformations, in: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, ACM, pp. 439–448.

[10] H. Fang, J. Hart, Detail preserving shape deformation in image editing, ACM Transactions on Graphics (TOG) 26 (2007) 12–es.

[11] H. Shungang, L. Shaoshuai, L. Xiaoxiao, Image deformation based on cubic splines and moving least squares, in: Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on, IEEE, pp. 1–4.

[12] T. Pal, A. Nutbourne, Two-dimensional curve synthesis using linear curvature elements, Computer-Aided Design 9 (1977) 121–134.

[13] A. Schechter, Synthesis of 2d curves by blending piecewise linear curvature profiles, Computer-Aided Design 10 (1978) 8 – 18.

[14] J. Stoer, Curve fitting with clothoidal splines, Journal of Research of the Notional Bureau of Standards 87 (1982) 317.

[15] D. Meek, D. Walton, Clothoid spline transition spirals, mathematics of computation 59 (1992) 117–133.

[16] T. Davis, Total least-squares spiral curve fitting, Journal of surveying engineering 125 (1999) 159–176.

[17] H. Späth, One dimensional spline interpolation algorithms, AK Peters, Ltd., 1995.

[18] T. Harada, F. Yoshimoto, Automatic curve fairing system using visual languages, in: Information Visualisation, 2001. Proceedings. Fifth International Conference on, IEEE, pp. 53–62.

[19] J. McCrae, K. Singh, Sketching piecewise clothoid curves, Computers & Graphics 33 (2009) 452–461.

[20] I. Baran, J. Lehtinen, J. Popović, Sketching clothoid splines using shortest paths, in: Computer Graphics Forum, volume 29, Wiley Online Library, pp. 655–664.

[21] G. Orbay, L. B. Kara, Beautification of design sketches using trainable stroke clustering and curve fitting, IEEE Transactions on Visualization and Computer Graphics 99 (2010).

[22] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, International Journal of Computer Vision 1 (1988) 321–331.

[23] P. Bao, L. Zhang, X. Wu, Canny edge detection enhancement by scale multiplication, IEEE transactions on pattern analysis and machine intelligence 27 (2005) 1485–1490.

[24] T. Harada, F. Yoshimoto, M. Moriyama, An aesthetic curve in the field of industrial design, in: Visual Languages, 1999. Proceedings. 1999 IEEE Symposium on, IEEE, pp. 38–47.

[25] K. Miura, J. Sone, A. Yamashita, T. Kaneko, Derivation of a general formula of aesthetic curves, in: In: Proceedings of the Eighth International Conference on Humans and Computers, pp. 166–171.

[26] K. Miura, A general equation of aesthetic curves and its self-affinity, Computer-Aided Design and Applications 3 (2006) 457–464.

[27] K. Shoemake, T. Duff, Matrix animation and polar decomposition, in: Proceedings of the conference on Graphics interface, volume 92, Citeseer, pp. 258–264.

[28] C. Xu, J. Prince, Snakes, shapes, and gradient vector flow, Image Processing, IEEE Transactions on 7 (1998) 359 –369.

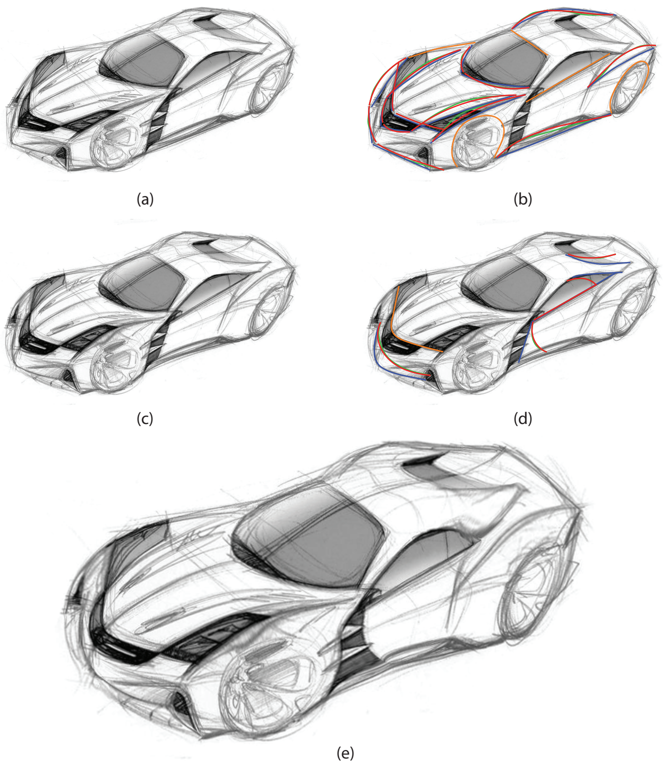

Figure A.15: (a) The original car sketch (b) The set of source and target curves that define the first set of intended modification (c) The intermediate result (d) The set of source and target curves that define the second set of intended modifications (e) The final result. The colors blue, green, and orange denote, the source, the beautified target, and the anchor curves, respectively.