EG UK Theory and Practice of Computer Graphics (2005)
L. Lever, M. McDerby (Editors)

Freehand 3D Curve Recognition and Oversketching

Li Han Raffaele De Amicis Giuseppe Conti

Graphitech via Dei Molini, 2, 38000, Trento (TN), Italy

Abstract

In the CAD/CAS field, the increasing domination of spline-based graphic objects has driven a great attention to methods focusing on the management of free-form curves. Based on the quick brainstorm illustration and stepwise refinement characteristics in conceptual designing stage, we present a method, which automatically reconstructs the designer's free-form 3D curve through recognizing his design "intention". This algorithm automatically extracts the relevant control points through corner detection and dynamic-threshold sampling mechanism; as a result the B-spline curve is approximately produced. Furthermore, considering the ambiguity of designer's intention during the conceptual designing, this redraw operation feature is implemented through the so-called "over-sketching". For this we introduce constrained length and tangent angle features, which supports fully free form 3D curve sketching, and it is capable of effectively smoothen transition interval. The method has been tested with various types of sketches, which are rendered in 3D scene environment. We further discuss the modification and its application to surfaces.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Free-Form Curve Recognition And Modification

1. Introduction

Freehand sketching is a very natural and powerful means of interpersonal communication. Especially in computer-aided design/styling (CAD/ CAS) fields most designers nowadays still prefer to work with pen and paper. This is due to the fact that freehand drawing is the most intuitive ones to capture designer's impulsive idea. Furthermore, this natural drawing feature lets designer more fully concentrate on the design process.

Recently, increasing domination by Spline-based graphic objects in CAD has driven much attention to create high degree free-form curve and surface. This further requires more intuitive tools which can assist designer to implement such creative designing works.

Unfortunately current Computer Aided Styling (CAS) softwares are still in general not simple to use for designers but require long adaptation training and an understanding of the math behind the geometries to modify curves and surfaces. Therefore, they are not suitable for the conceptual designing stage, where requires less computing but fast and various illustrations by sketching.

The earlier works about curve [FB93, Bar81, PT95] [BBB*93, GA98, QWJ01] tend to modify parameters, such as spline control points or knot values and sophisticated curvature, tangent constraints; consequently they affect the curve's shape. Although the representation of curves matching the constraints of the objects provides perfect shape control and useful geometric properties, e.g. C2 continuity where needed and easy computation of bounding surface or volume, the users usually have some background in mathematics enabling them to master easily the peculiarities of the interaction technique.

Significant achievements in user interaction for free hand drawing were made by [Bau94] [MKB02, CM99], they proposed methods which designers can freely modify their drawings by simple touch-and-replace or projection techniques, however the restrictions of them are the edited representations are based on piecewise linear B-spline, and they do not effectively resolve the smooth in transition interval. In [TF04] they proposed a resolution to the smooth of transition interval; however it cannot guarantee the effectivity in the case of loop curves.

Indeed when user is freehand sketching, the result is not a basic B-spline or Bezier curve, it is arbitrary object, and so first we should consider how to fit data into some basic curves. In [BC90] they propose the real-time interaction which is able to reduce the data incrementally using knot removal method, and in [FMR*02]the authors adopt the ad-hoc curve splitting method to approximate B-spline in VR environment, but it involves in large computation and the complex parameters are not suitable for supporting the intuitive post-processing in conceptual designing stage.

From the above reviews, we can find two main deficiencies of the existing methods for free-form curve sketching.

- Strict restrictions on the cumbersome geometric parameters representation and mathematic computation.
- Lack of comprehensive way to analyse the imprecise data from free-form 3D curve sketching and to implement the post-refinement such as intuitive local modification.

In this paper, we present an innovative method, which is capable of automatically approximating imprecise free-drawing curve to adaptive B spline base on dynamic-threshold sampling mechanism. We further propose stroke over-sketching algorithm to implement stroke-replace local modification, it supports fully free form curve sketching. Furthermore we improve the smoothness of transition interval through the definition dynamic scale factor and the constrained tangent angle features.

The rest paper is organized as following: In section 2 we present previous works and the overview of our system. Section 3 introduces free form 3D curve sketching and adaptive B-spline approximation algorithm. In section 4 we will detail the redrawing feature for local modification and present the novel method to deal with the transition interval. Section 5 concludes with a summary and describes the directions of future work.

2. Previous works and the overview of our system

2.1 Previous works

Many curve researches have been developed in the past. In [FB93] the authors find a new way to directly manipulate B-spline curves, using constrained normal, curvature and knots vector, which leaves the burden to understand the relation of geometric parameters to the users. The work in [Bar81] proposed easy edition of B-spline curves by cutting and sketching control polygons. It still means additional work to edit control points, rather than curves. In [PT95] they use the knot removal technique to adjust curve shape, however it increases computation time and complexity of curve's mathematic representation. In [Kar99] they present an efficient technique for approximating a planar parametric curve by a small set of elliptic arcs; they also introduce a simple approach to smooth replaced sections of design curves with sections of the French curve. This system is of great value in simplifying and neatening curve, but it would involve in a number of computation, especially it is not suitable for the users who without the professional experience.

The usual approach to approximation curve uses a least squares formulation, and it expresses the fairness of the final result based on parametric B-splines. But it is difficult to estimate the parameters. In [PL02] the authors present the active curve or surface adapts to the model shape to be approximated in an optimization algorithm. This is built up with help of local quadratic approximants of the squared distance function of the model shape and an internal energy; as a result it improves the smooth and regular effect. Although it avoids the parameterization problem, it also involves in too complex mathematic knowledge to understand the relationship between B-spline curves and B-spline surface.

The significant free form drawing system is made in [Bau94], it proposes a freely modification method by pen strokes instead of controlling the knots and control points of curves. One restriction in this approach is that it uses off-line spline curves, the edited representations are based on piecewise linear curves, it cannot really implement free-hand drawing mode. The improved works in [MKB02] presents simple touch-and-replace technique to edit 2D and to generate 3D curves; and they introduce auxiliary surfaces sculpting that allow for a reliable interpretation of users' pen-strokes in 3D. However the editing operation is implemented by interpolating, it cannot guarantee the smooth of surface. In addition, it is lack of degree of freedom restricted by the auxiliary surface and the poor visualization for sculpting operation.

The work in [CM99] introduce that the shape of a 3D curve is determined by its image-space projection and its shadow. This system is well suitable for application that requires fast specification of approximation 3D curves. But the more precise curves might not benefit from this technique.

Indeed when user is free hand sketching, the result is arbitrary object; we have to fit data into some basic curves, such as B-spline or Bezier. Most previous data fitting schemes, including sketching algorithms, adopt one of two approaches: One starts with an initial approximation with a few degree of freedom and splits it into more piecewise polynomial segments based on some pre-specified criteria. That process iterates until sufficient accuracy is obtained. The scheme presumes that all the data is present; Other ("greedy") schemes work from the initial end and use a single polynomial to represent the data until that can no longer be done and then use piecewise Hermit/Bezier ideas to estimate tangents and continue with another piece.

In [BC90] they propose a new data reduction algorithm, which is based on LM algorithm incrementally removing the knot vector using "window size". Although it constructs the curve in interactive real-time hand drawing, they do not analysis the effective window size for different situations, it also involves in excessive computation. In [FMR*02]it translates conceptual sketch strokes into a suitable B-spline representation with a three-step method, and uses the virtual reality (VR) techniques, instead of traditional two-dimension devices. However it cannot guarantee effectivity in general cases.

2.2 The overview of our system

The whole process of our system is divided into two stages: imprecise free-form 3D curve approximation and post-refinement. The former will be performed immediately after each stroke is finished, the result will be displayed as a rapid feedback. The adoption of optimal sampling mechanism effectively improves the approximation result.

Post-refinement will be implemented by intuitive stoke replacement operation which is so-called “oversketching”, we present a novel resolution to the smooth in transition interval using the dynamic scale factor and the constrained length and tangent features

Our system is easy to use. It does not put any particular constraint on the manner of drawing sketches. Users can use any pen-like input device or even a mouse to draw freely. Our system will record the strokes, which are defined as a series of coordinates visited by the pen between a single pair of mouse-down and mouse-up evens. Besides coordinates, each stroke point associates with a time stamp to get corresponding speed features which is important for extracting geometric features.

The method has been tested with various types of sketches, which are rendered in 3D scene environment. Furthermore we employ this method to free form surface.

3. Free-form 3D curve sketching and approximation

Our approach supports freehand drawing, automatically analysising and adapting into B-spline when sketching curves directly in 3D space. Three steps are followed:

- Preprocess: Filtering and removing the redundant data and noise.
- Sampling:
    1. The introduction of the corner detection algorithm based on the analysis of speed, curvature and direction features splits the curve into several parts.
    2. We adopt dynamic-threshold method which combines arc length and speed features to obtain optimal sampling points.
- Approximation: The algorithm of approximation translates all the sampling data into adaptive B-spline.

3.1 The input data filtering

An unfortunate property of data input devices, especially digitizing tablets, is that the digitized data stream may contain a number of duplicate data values. In addition, data values may simply be more closely spaced than necessary in consideration of the allowable error. Much of this is due to the relationship between the speed of the user’s hand movement, and the sampling rate of the input device. Also hand sketching a curve is a continuous process, whereas sampling the motion is necessarily discrete.

Here we adopt two-step filtering method to remedy these problems. First, as the data is received from the input device, any duplicate data value is removed. Second, if the input data are within some small distance of each other, the data values are filtered with a Gaussian filter of kernel σ = 3, where we apply a 3×3 mean filter (see Figure 1 a), which is simply to replace each pixel value with the neighborhood value. This effectively removes much of the noise present in the digitized data (see Figure 1 b).

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

(a) (b)

Figure1: *a) 3×3 averaging filter b) data filtering*

3.2 Dynamic-threshold based sampling method

During the freehand 3D curve drawing, the corner points are very important information for keeping the curve shape. Here we adopt two parameters to detect corners: *direction and curvature.*

When free-form curve sketching, the direction and curvature for each point are automatically computed (see formula 1): where d_n and C_n represent the direction and curvature of the n-th stroke point respectively, k is a small integer defining the neighborhood size around the n-th point, and $D_{(n-k,\, n+k)}$ stands for the path length between the $(n-k)$-th and $(n+k)$-th stroke points. We set k to 2 empirically as a tradeoff between the suppression of noise and the sensitivity of vertex detection. The function “φ” shifts its angle parameter from $-\pi$ to π (see Figure 2).

$$d_n = \arctan\left(\frac{y_{n+1} - y_n}{x_{n+1} - x_n}\right)$$

$$C_n = \frac{\left|\sum_{i=n-k}^{n+k-1} \varphi(d_{i+1} - d_i)\right|}{D(n-k, n+k)} \qquad (1)$$

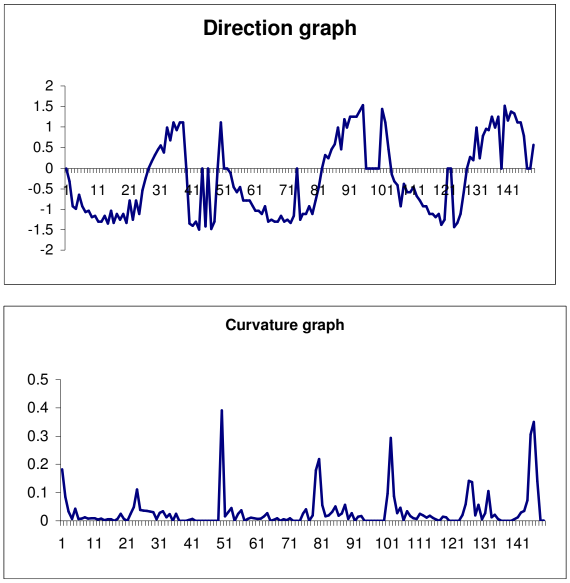


Figure 2: *an example curve with its direction graph and curvature graph*

In the Figure 2 we can find the direction graph where the direction angle is shifted from $-\pi/2$ to $\pi/2$ has little value for analysis. However in curvature graph we can easily judge the corner point region that are necessary to preserve the shape.

In fact the freehand sketching is an unsteady movement. When the speed is low, it is easy to obtain much more sampling points and to understand the user's intension; on the contrary, when the speed is high it is good to get smooth shape but the sampling points are few, it would result in the difficulty for post-processing. Therefore, we combine two cases to adaptively obtain sampling points. In a similar manner to the approach proposed in [QWJ01], we discriminate them by using different sampling threshold; Aiming at the usual low speed in corner region and high speed in other parts, we combine the arc-length and speed features to get optimal sampling points.

$$m_i = round(2 \times \frac{S_{avg}}{S_i} + 0.5) \qquad (2)$$

$$m_i = 7 \times \frac{S_{avg}}{S_i} \qquad (3)$$

Where m is the adaptive length threshold for sampling, and m_i is the distance from (i-1)-th point to i-th point. S_{avg} is the average sketching speed which is defined by the arc-length and time stamp, and S_i is the speed at i-th point, the purpose of m is to avoid picking too close points. Considering the fast speed in non-corner region we employ formula 3 to control the distance between points; on the other hand we use formula 2 to get sampling points in corner region where it is slow speed.

3.3 Adaptive B-spline approximation

The final result of the complete algorithm is to provide a set of mathematical representation of the curves keeping them simple for further modification. This is usually done by interpolating all the data points or approximating the data to get desirable curves.

Due to it is well-known that hand-sketching is susceptible to the unsteady and imprecise hand movements of the designers, interpolating all the data generally suffers from excessive undulations, especially when the number of data points increasing or maintaining high continuity, it will result in sharp "wiggles".

Therefore, we adopt the Carl de Boor's least squares approximation method to repraramleterize sampled points, where the B-spline is specified by its non-decreasing knots sequence and sampled control points sequence. (See formula 4, 5, 6).

$$C(u) = \sum_{i=0}^{n} B_i P_i \qquad (4)$$

$$B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}, 0 \le i \le n \qquad (5)$$

Where P_i represents the sampled control points, n is the number of control points, and B_i are cubic B-spline basic functions over a knot vector u, its value ranges from 0.0 to 1.0 (see formula 6).

$$U = \{u_0, u_1, u_2 \ldots u_{n+1}, u_{n+2}, u_{n+3}, u_{n+4}\} \qquad (6)$$

We suppose the B-Spline curve is a piecewise polynomial or rational curve. The knots are the parameters of junction points between two pieces. The multiplicity *Mult (i)* of the knot *Knot (i)* of the B-Spline curve is related to the degree of continuity of the curve at the knot *Knot (i),* which is equal to *Degree - Mult (i)*, where *Degree* is the degree of the B-Spline curve.

Here we decide to use cubic splines to keep C^2 continuity and the knots sequence distributed with multiplicity equal to order at extremities; based on the optimal sampled points we can approximate the adaptive B-spline (see Figure 3).

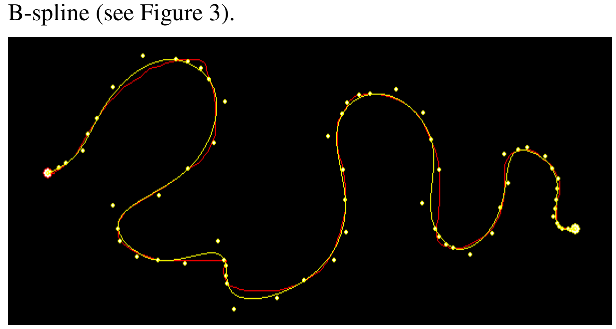


Figure 3: *free form curve and approximated B-spline*
(The red one is the freeform sketching;
The yellow one is approximate B-spline)

4. Local modification based on simple oversketching

Due to the stepwise refinement characteristic in conceptual designing stage, intuitive and natural modification operation is the most important feature for the fast illustrations of designers' ideas; therefore, we adopt simple redrawing operation by the so-called stroke "oversketching" (see Figure 4).

For this, there are general two modification methods, one is the replacement of control points, and another one is the replacement of all sampling points. Due to we already get optimal control points during the approximation procedure, and in order to improve the smooth and flexible characteristics of curves, we will detail the local modification process based on the replacement of control points.

Figure 4: *local modifications based on oversketching*

During the modification, the important task is to find the replacement part in the original curve; we adopt the projection method to find the nearest points from the start point and the end point in the objective curve to the original curve (see Figure 5). Where *C1 (u)* represents the original curve, and *C2 (u)* is the new stroke that would be approximated as objective curve. P_s' and P_e' are the projections of P_s and P_e, if we directly replace the $\overline{P_s'P_e'}$ with *C2 (u)*, it would produce sharp breaks.

To avoid that, the curve is smoothening in transition intervals as described in the next part. Then the new revised curve will include three parts, $C(u)=C_{11}(a_1, a_2 \ldots, a_m) + C_2(b_1,b_2,\ldots,b_n)+C_{12}(c_1,c_2,\ldots c_r)$, where C_{11} and C_{12} are the two parts of the original curve, C_2 is the objective curve, a_i, b_i, c_i mean a series of control points.

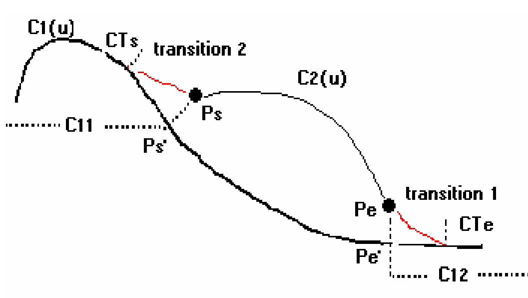


Figure 5: *local modification*

The whole modification process can be illustrated as following:

- Calculating the projections: As illustrated in Figure 5 we can obtain the projection points P_s' and P_e' from P_s and P_e. Furthermore, we Locate the nearest control points CT_s and CT_e in *C1 (u)*.
- Replacing: The part curve $\overline{P_s'P_e'}$ in original curve *C1 (u)* will be replaced by objective curve *C2 (u)*.
- Defining the transition intervals: we introduce dynamic scale factor and tangent features to adjust the position of CT_s and CT_e in order to obtain the optimal transition intervals.
- Re-Parameterzing: Based on corresponding control points we re-parameterize the knots sequence and get a new B-spline curve.

4.1 The resolution to transition interval

To avoid the sharp beak in the curve, we adopt two methods to improve this smoothness.

- Dynamic scale factor
- The constrained length Δd and tangent angel $\Delta \varphi$ for the definition of transition interval

In [TF04] the author proposed a similar scale feature to resolve the smooth of transition parts; however the adoption of interpolation method results in the whole curve sharp undulation. It is ineffective in the case of loop curves or more complex curves.

We further improve this flexibility through approximation method, which is based on a series of optimization control points that we already obtained during the recognition process.

In our method the scale is used for weighting the degree of approximation to the objective curve; it is defined as float type where the value ranges from 0 to 1. When it is 0 the original curve will not be modified and if it is 1 the new curve will follow exactly all the control points in the objective curve.

We calculate all the projection distances from the control points in *C2 (u)* to *C1 (u)* (see Figure 6). We suppose y_t is the one control point in *C2 (u),* the projection distance is d_t; then we can produce the new control point y_t' through a scale (see formula 7).

$$y'_i = y_i - (1\text{-scale}) * d_i \qquad (7)$$

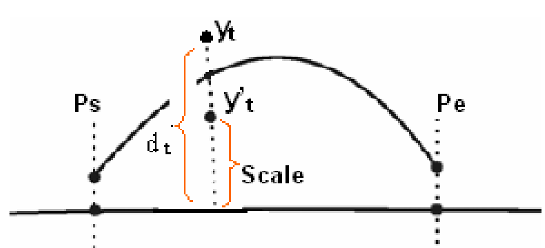


Figure 6: *Scale for modification*

Even if the scale factor adjusts the degree of approximation to objective curve, however when the distance from P_s *to* CT_s or from P_e to CT_e, is too closer, the hard break would be still existed. So we introduce the constraints of length Δd and tangent angle $\Delta \varphi$ to obtain the suitable control point CT_s and CT_e in *C1 (u),* consequently we can decide the optimal transition region.

$$\overline{M} \le \Delta d \le 1.5\,\overline{M} \qquad (8)$$

$$\overline{M} = \frac{\sum_{i=1}^{n} round(2\times\frac{S_{avg}}{S_i}+0.5)}{n} \qquad (9)$$

The length Δd is the distance of $\|\overline{CT_s P_s}\|$ or $\|\overline{P_e CT_e}\|$. It is empirically defined as a tradeoff that its value should be satisfied with between $\overline{M}$ and 1.5 $\overline{M}$ (see formula 8), $\overline{M}$ is the mean distance between different neighbor control points in original curve *C1 (u)*. According to formula 9 we can deduce the value of $\overline{M}$.

We further refine the transition interval by adopting the constrained feature $\Delta \varphi$. $\Delta \varphi$ is used for evaluating the difference of tangent angle between $\overline{P_s CT_s}$ and $\overline{CT_s CT_{s-1}}$ or $\overline{P_e CT_e}$ and $\overline{CT_e CT_{e-1}}$. For instance: we suppose P_s (x_0, y_0) is the start point of objective curve *C2 (u)*, the corresponding nearest control point is CT_s (x_s, y_s) in *C1 (u)* and CT_{s-1} (x_{s-1}, y_{s-1}) is the previous control point, as described in formula 10, we can obtain the tangent angle difference $\Delta \varphi$.

$$d_1 = \arctan(\frac{y_0 - y_s}{x_0 - x_s})\ ; \qquad d_2 = \arctan(\frac{y_s - y_{s-1}}{x_s - x_{s-1}})\ ;$$

$$0 < \Delta \varphi = |d_1 - d_2| < \pi/2 \qquad (10)$$

During the whole process of defining of transition intervals, we firstly use the length threshold Δd roughly calculate transition region; secondly we adopt the least deference of tangent angle to obtain the exact control points.

The combination between scale factor and the constraints of length and tangent angle provides the effective manner to improve the smooth feature in transition intervals; furthermore this method is suitable for fully free form 3d curve sketching (see Figure 7 - Figure 10).

4.2 Surface modification

We further employ this method to free-form surface where the surface is constructed through freehand 3D curves sketching. We adopt the *GeomFill* function in Open Cascade; in this way we can create surfaces either from boundary curves or respecting curve and point constraints.

The process follows four steps:

STEP1: Free form 3D curves sketching and automatically approximating into B-spline curves based on our algorithm described in section 2.

STEP2: Using *GeomFill* function to create surface where the curves work as boundary.

STEP3: Adopting our local modification approach stated in section 3 to revise surface's boundary.

STEP4: Creating new surface based on the revised boundary.

A large number experiment verified the effectivity of this intuitive operation. Consequently, this further extends the field of surface modeling by using the approximation free form 3D curve sketching (see Figure 11).

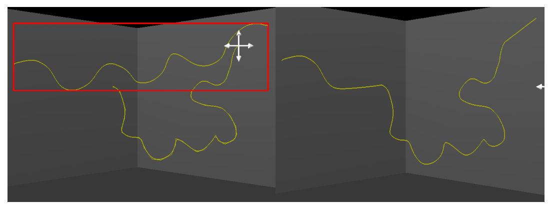


Figure 7: *Scale =1*

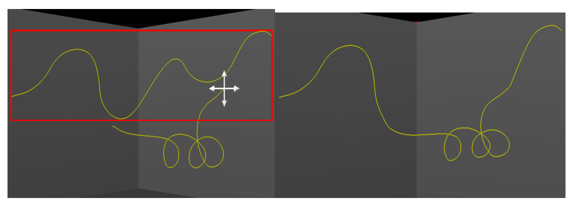


Figure 8: loop *curve local modification*

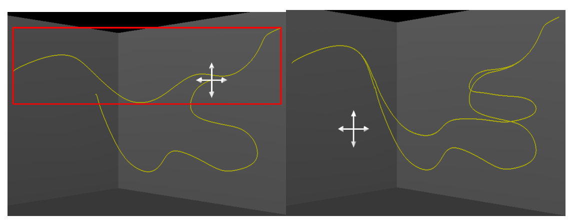


Figure 9: *the upper one is generated from Scale=0.5*

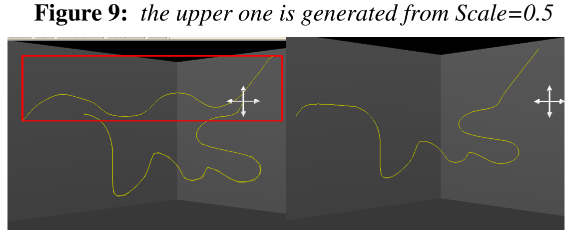
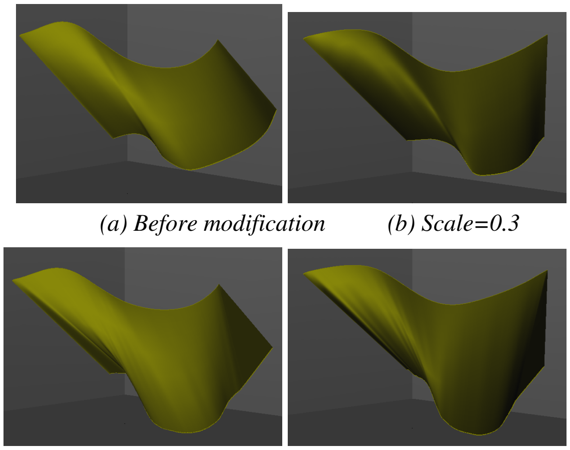


Figure 10: *the constrained length 0.8* $\le \Delta d \le$ 1.2



(a) Before modification *(b) Scale=0.3*

(c) Scale=0.8 *(d) Scale=1*

Figure 11: *modifying the boundary of surface based on oversketching (1.2 ≤ Δ d ≤ 1.8)*

5. Conclusion and future work

In this paper we presented a comprehensive technique for any free-form 3D curve sketching and intuitive local modification, it aims at achieving the stepwise refinement process which is typical of the early stages of the design process. This method presented features dynamic-threshold based sampling mechanism that is capable of obtains optimal sampling data and adaptively approximating into B-spline. In particularly, we propose an innovative local modification algorithm, which implements simple stroke oversketching operation by approximating correspond control points. The introduction of dynamic scale factor and the constrained length and tangent angle features improves the smooth effect of modified curve. A series of experiments indicate the effectivity of this algorithm. In a sense we achieve a reasonable compromise between real-word design tools with computer-aided tools. This technique will provide the strong basis for industrial designing such as car styling.

In future we will further discuss the interior surface deformation based on simple stroke oversketching. In particular we will develop much more interactive mode for free-form surface modeling and manipulating. Furthermore, we will apply them to the creative car styling area.

Acknowledgements This research presented in this paper is part of the EU project "IMPROVE".

6. References

[FB93] Fowler B. and Bartels R.: Constraint-based curve manipulation. *In IEEE Computer Graphics and Applications* (Sept. 1993), 43–49.

[Bar81] Barsky B.A.: The Beta -Spline: A local representation based on shape parameters and fundamental geometric measures, *PhD thesis, University of Utah.*(1981).

[PT95] Piegl L. and Tiller W.: The Nurbs Book. *Springer Verlag*, (1995).

[GA98] Grimm Cindy and Ayers Matthew: A framework for synchronized editing of multiple curve representations. *In EUROGRAPHICS '98,* (1998), C-31 – C-40.

[Kar99] Karan Singh: Interactive curve design using digital French curves, *Proceedings of the 1999 symposium on Interactive 3D graphics* (April. 1999).23-30,

[PL02] Pottmann Helmut, Leopoldseder Stefan: Approximation with Active B-Spline Curves and Surfaces, *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications* (Oct. 2002) 8.

[Bau94] Baudel T.: A Mark-based Interaction Technique for Free-Hand Drawing. *Proceedings of the ACM UIST'94 Conference on User Interface Software and Technology*, 185-192.

[MKB02] Michalik P., Kim D., Bruderlin B.: Sketch- and Constraint-based Design of B-spline Surfaces. *In Proceedings of International Conference on Solid Modeling* (2002).

[CM99] Cohen Jonathan M., Markosian Lee: An interface for sketching 3D curves. *ACM I3DG 1999 Symposium on Interactive 3D Graphics.* 17-21.

[FMR*02] Fiorentino M., Monno G., Renzulli P.A., Antonio E. Uva: 3D Sketch Stroke Segmentation and Fitting in Virtual Reality.*International conference on the Computer Graphics and Vision, GRAPHICON* (Sept. 5 - 10, 2003), Moscow, Russia.

[BC90] Banks M., Cohen E.: Real-time Spline Curves from Interactively Sketched Data. *Proc. SIGGRAPH Symposium on Interactive 3D Graphics* (1990), 99-107

[BBB*93] Bartels R. H., Beatty J.C., Booth K. S., Bosch E.G., Jolicoeur P.: Experimental Comparison of Splines using the Shape-Matchiag Paradigm, *ACM Transactions on Graphics. Vol. 12 (3),* (1993), 179-208.

[QWJ01] Qin, Sheng-Feng, Wright David K., and Jordanov Ivan N.: On-line segmentation of freehand sketches by knowledge-based nonlinear thresholding operations. *Pattern Recognition, vol.34* (number 10 2001), 1885-1893.

[TF04] Timao Fleisch, Florian Rechel: constraint stroke-based oversketching for 3D curves . *EUPROGRAPHICS workshop on sketch-based interfaces and modeling* (2004), 161-165.