

Constrained Shape Modification of B-Spline Curves

Mukul Tuli¹, N. Venkata Reddy¹ and Anupam Saxena¹

¹Indian Institute of Technology Kanpur, mukult@iitk.ac.in, nvr@iitk.ac.in, anupams@iitk.ac.in

ABSTRACT

This paper proposes a methodology to change the shape of a *B-Spline* curve locally. Present algorithm modifies the existing *B-Spline* curve and makes it pass through a given point P, constrained to lie in region R, with limited deviation from the original curve. Proposed methodology splits the curve into atmost three segments based on the curvature of the curve and then modifies only one segment using knot insertion and control point inclusion.

Keywords: *B-Spline* curve; Knot Insertion; Curve splitting.

1. INTRODUCTION

A *B-Spline* curve $C(u)$ is defined by control points (also known as de-Boor points) P_i and the knot vector $U = \{u_1, u_2, \dots, u_{p+n}\}$ as

$$C(u) = \sum_{i=1}^n P_i N_{i,p}(u) \quad (1)$$

where p is the degree of the curve, n is the number of control points and $N_{i,p}(u)$ are the spline basis functions [4] defined over a knot vector U .

During the past few decades, *B-spline* curves have gained popularity in shape modeling and geometric design [1]. Beyond creating these types of curves, modification of the existing ones is also of great importance. *B-spline* curves offer a unified mathematical formulation for representing not only free-form curves, but also standard analytical shapes. Through the manipulation of control points, degree and knots, a user can design variety of shapes using *B-spline* formulations. In a system that uses *B-Spline* curves as shape design tool, one can implement many algorithms such as position and derivative evaluations, knot insertion [2], knot deletion and degree elevation [4]. Based on these fundamental algorithms, one can perform geometric modeling and processing.

B-Spline follows a representation based on control points and knots, and weights in case of rational *B-Spline* curves. A designer can modify a local curve segment by iteratively moving a control point, changing the knots or modifying the weights associated with the control points. To this end, Piegl [3] presented a method for modifying shape of a rational *B-Spline* curve by relocating points on the curve. The modification is realized by repositioning one or more control points. Piegl and Tiller [4] discussed in detail curve wrapping, flattening, bending and twisting using the same approach. In addition, Au and Yuen [5] investigated how the shape of a NURBS curve changes when the weights and location of the control points are modified simultaneously. A drawback that the above mentioned methods have is that they are not automatic. A designer should predefine the control point to be shifted or the corresponding weight to be changed to provide the desired shape modification.

Working on the interactive scheme for curve modification, Fowler and Barter [6] had presented constrained based manipulation of curve of arbitrary degree. A direct modification of free form curve by displacement functions was proposed in [7], which comprises of degree elevation and control point repositioning.

Besides the most basic tool for shape modification, control point or weight modification, one can work on knot alteration. Juhasz and Hoffmann [8] showed that modification of a knot of *B-Spline* curve of degree p generates a one parameter family of curves. The family has an envelop which is also a *B-Spline* curve with the same control polygon and degree $p-1$. The theoretical results have been applied in [8] to modify the shape of a cubic *B-Spline* curve. The

approach used determines three consecutive knots by fitting a curve of a parabolic family based on the desired constraints. The procedure enables a local shape modification but is, however, restricted to cubic *B-Spline* curves only.

Some other aspects of knot modification have also been studied by Lyche and Morken [9], where the effect of knot variation is examined from numerical point of view. The choice of knot value in curve approximation and interpolation has also been investigated in several papers, the recently published [10] and references therein. There are not many attempts in the area of knot alteration to get the desired shape change. The main reason for this is the non intuitive nature of curve change by knot alteration. The problem arises because with each different placement of knots one can derive a new set of control points, which can exactly get the same curve [11]. This problem is further compounded by the fact that the estimation of knot value is a non linear process. Czuczor et al. [12] and Mamic et al. [13] used optimization procedures to overcome the problem faced in knot vector alteration. The use of optimization is computationally expensive. Present work is an attempt to find the *B-Spline* curve with desired change without any iteration involved.

The most effective shape control tool of *B-Spline* curves is control point repositioning. A remarkable advantage of knot modification over control point repositioning is that the modified curve always remains in the region determined by the convex hull of the original curve. Thus, the knot based method is preferred if one wants the curve in that region or when fine tuning in the curve is required [8].

Present work involves modifying the shape of an existing *B-Spline* curve with the shape preserving constraint and less deviation from the original curve. The approach uses both knot vector alteration by introducing new knots using knot insertion [14] and control point inclusion. Unlike the approach of fitting a *B-Spline* curve to the given data [15], this method works at making the curve change locally. The *B-Spline* curve fitting problem is to produce a *B-Spline* curve to approximate a target curve within a pre-specified tolerance. This fitting technique uses an energy function and then minimizes it to find the new position of the control points. On the contrary, the present work modifies the curve locally, by splitting it into parts and then achieves the desired shape change without any optimization involved.

2. PROBLEM DESCRIPTION

The main aim of the present work is to make a *B-Spline* curve $C(u)$ pass through a prescribed point \mathbf{P} (Fig. 1.) which lies in a constrained region R (described later in this section). The other objective of this work is to preserve the curve shape and to have limited deviation from the original curve. The shape of the curve is assumed to be preserved if the modified curve has the same number of feature points as the original curve.

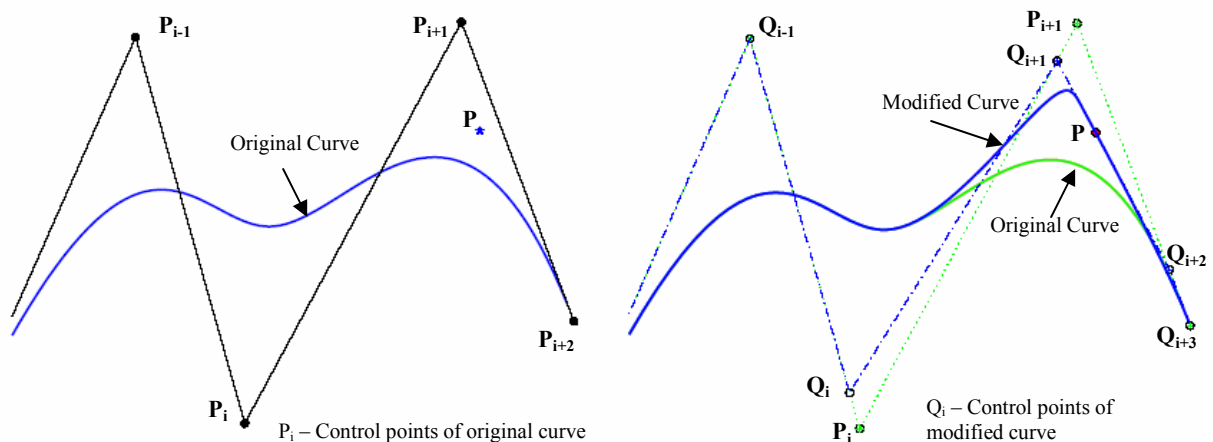


Fig.1. Problem definition, original curve and the modified curve.

Every curve has certain features that define the shape of that curve, namely, maxima and minima of curvature and sudden changes in the curvatures. These points are defined in the present work as feature points \mathbf{F}_i . To find these feature points, discrete curvatures [16] at regular interval on the curve are evaluated. These feature points are identified using fine and rough checking technique proposed by Liu et al. [16]. Feature points are intrinsic to the curve

and are used in section 3 to find the ends of the split segment. To have limited deviation from the original curve the shape modification is performed locally, only in the curve segment between the split ends (section 3).

The proposed algorithm for constrained shape modification works when point \mathbf{P} is present within a constrained region R as shown in Fig.2(b). To modify the curve, the essential step is to find the value of knot to be inserted in the existing knot vector using the concept of knot insertion and mapping knots on the control polyline as discussed by Sederberg [14].

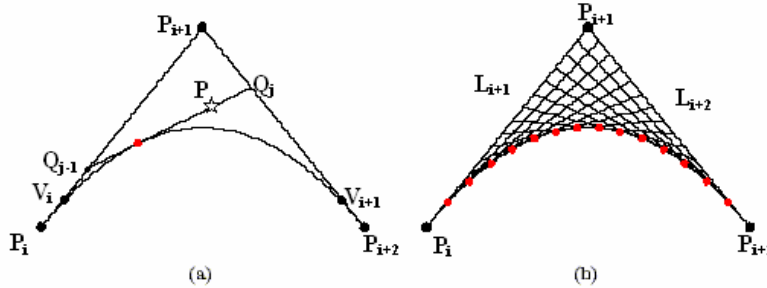


Fig. 2(a). Finding the knot value to be inserted, 2(b). Hashed region R.

2.1 Admissible Region R for Point P

Let $\mathbf{P}_i, \mathbf{P}_{i+1}, \mathbf{P}_{i+2}$ be three consecutive control points of a degree p *B-Spline* curve (Eqn. (1)). Consider knots $u_i, u_{i+1}, u_{i+p}, u_{i+p+1} \in U$. Define

$$V_i = \frac{(u_{i+1} - u_i)P_{i+1} + (u_{i+p} - u_{i+1})P_i}{u_{i+p} - u_i} \quad \text{and} \quad V_{i+1} = \frac{(u_{i+p+1} - u_{i+p})P_{i+1} + (u_{i+p} - u_{i+1})P_{i+2}}{u_{i+p+1} - u_{i+1}} \quad (2)$$

to be the points on the polygonal leg $\mathbf{L}_{i+1}(\mathbf{P}_{i+1}, \mathbf{P}_i)$ and $\mathbf{L}_{i+2}(\mathbf{P}_{i+2}, \mathbf{P}_{i+1})$, as shown in Fig.2. Here \mathbf{V}_i and \mathbf{V}_{i+1} correspond to the mapped positions of knot u_{i+p} on \mathbf{L}_{i+1} and u_{i+p+1} on \mathbf{L}_{i+2} respectively. The region R has three boundaries, two formed by the control polyline and one by the parabola with starting and ending points as \mathbf{V}_i and \mathbf{V}_{i+1} .

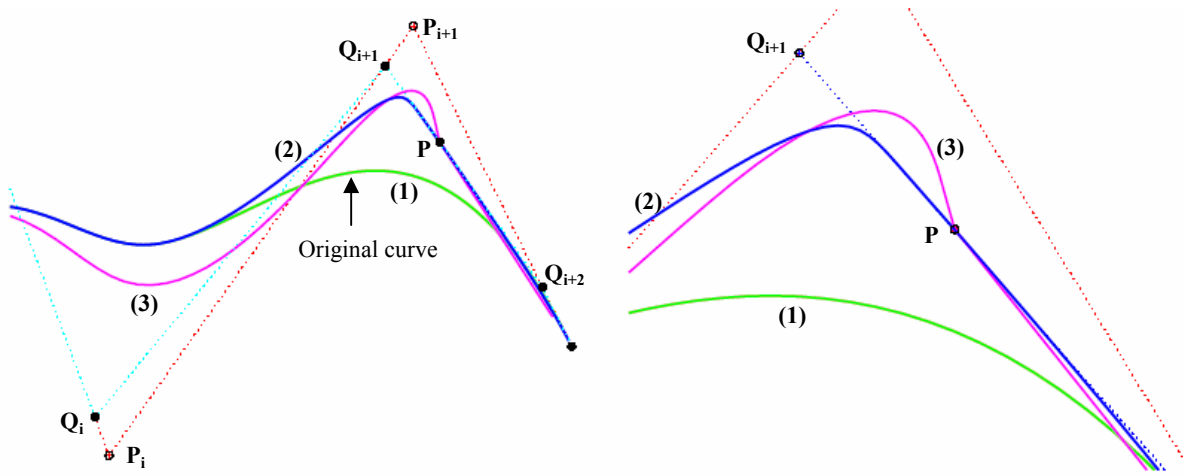
2.2 Algorithm for Constrained Shape Modification

The given *B-Spline* curve is made to pass through a point $\mathbf{P} \subseteq R$ by inserting a knot u in U , such that one of the newly formed control polyline $\{\mathbf{Q}_{j-1}, \mathbf{Q}_j\}$ [4], passes through point \mathbf{P} (Fig.2(a)). Note that \mathbf{Q}_{j-1} and \mathbf{Q}_j are the mapped positions of knot u on the polygonal legs \mathbf{L}_{i+1} and \mathbf{L}_{i+2} respectively. The steps involved in constrained shape modification are:

- The value of knot u is found such that $\mathbf{P} = \alpha \mathbf{Q}_{j-1}(u) + (1 - \alpha) \mathbf{Q}_j(u)$ where $\alpha \in [0, 1]$.
- Using knot insertion [4], new control points \mathbf{Q}_j are found.
- Point \mathbf{P} is included in the control point set as a control point of multiplicity $p-2$.

The use of knot insertion reduces the size of the convex hull and brings it nearer to the curve keeping the curve unchanged. The reduction in convex hull size limits the local deviation of the curve in a smaller region (Fig.1.). A *B-Spline* curve is monotone if the control polygon is monotone and is convex if it is convex. The present algorithm finds a control polygon by inserting knot u , which maintains the convexity. Next, inclusion of point \mathbf{P} as control point does not change the convexity of newly formed control polygon, since the polygon remains the same with the reduction in convex hull size. As there is no change in the convexity of the polygon, the shape of the modified curve is preserved.

Procedure of finding new control points, a set of \mathbf{Q}_j such that \mathbf{P} lies on the polyline and including \mathbf{P} as a control point of multiplicity $p-2$ makes one the newly formed polyline tangent to the modified curve at \mathbf{P} . This produces a smooth curve. It can be seen in Fig.3(a) that polyline $\mathbf{Q}_{i+1}\mathbf{Q}_{i+2}$ forms a tangent to the curve at \mathbf{P} . There will be a formation of



(a). Original curve (1), modified curve using knot insertion (2) and without knot insertion (3). (b) Formation of Kink in curve (3).
 Fig. 3. Constrained shape modification.

kink in the curve if the curve is modified by directly including P as a control point of multiplicity $p-2$ without knot insertion, as shown for curve (3) in Fig.3.(b).

3. DETERMINATION OF SPLIT SEGMENT

Modification of the *B-Spline* curve affects the shape of the curve in $p-1$ convex hull [4]. To limit the shape change to further smaller region, the curve is split into almost three segments depending on the positions of point P in space. The shape of the segment nearest to the point P is changed. Feature points defined in section 2 are used for the purpose of finding the appropriate positions of the split ends. An intrinsic local shape modification method is developed using these feature points.

3.1 Determination of Critical Feature Points

To determine the segment to be modified, the shortest distance of the point P from the original *B-Spline* curve is evaluated. Fig.4 shows a part of a *B-Spline* curve and point P . The feature points F_{i-3} to F_{i+2} are marked on the curve. Point Q^* denotes the point on the curve nearest to point P and let D be the distance between them. Let $d_{i,i+1}$ denote the length of the curve between two consecutive feature points, F_i and F_{i+1} . This length controls the placement of the feature points on the curve. It is clear that the curve will have frequent changes in curvature if $d_{i,i+1}$ is small as compared to a large value of $d_{i,i+1}$. Consider that point Q^* lies on the segment $F_i F_{i+1}$, the length of segment $F_i Q^*$ and $Q^* F_{i+1}$ are denoted as l_i and l_{i+1} respectively.

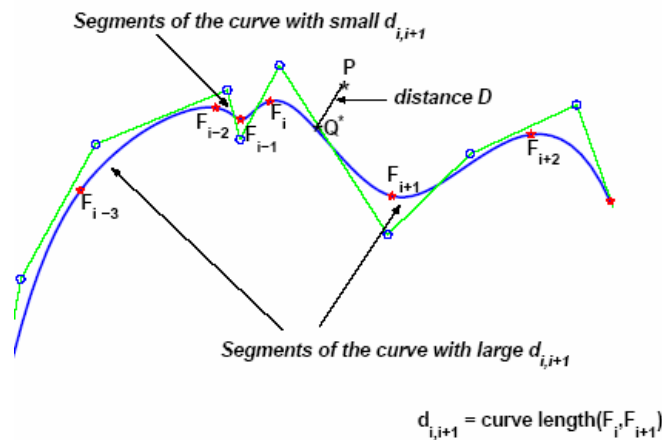


Fig. 4. Feature points on the curve.

Let

$$d_m = \min(l_i, l_{i+1}) \quad \text{and} \quad I_m = \begin{cases} i & l_i < l_{i+1} \\ i+1 & \text{otherwise} \end{cases} \quad (3)$$

Thus F_{I_m} denotes the feature point nearest to point \mathbf{P} . In order to preserve the shape of the curve, it is important to preserve the features in the neighborhood of the desired point \mathbf{P} , defined as critical features in this work. Thus F_{I_m} is one of the critical features. There could be a need to include more features as critical features to provide a smooth transition of the curve at the ends of the split segment. It can be observed in curve (1) of Fig.5 that inclusion of only one feature as critical feature may lead to an abrupt change in curve shape at split ends. Curve (2) also shown in the same figure, with three critical feature points, shows a smooth transition of the modified curve to the other segments at the split ends. In order to determine the other critical features, feature points present in the neighborhood of \mathbf{P} are identified using a threshold distance parameter $\tilde{a}(i)$ defined as

$$\tilde{a}(i) = \frac{\frac{3}{2}(d_{i-1,i} + d_{i,i+1})}{2 + \frac{\max(d_{i-1,i}, d_{i,i+1})}{\min(d_{i-1,i}, d_{i,i+1})}} \quad (4)$$

This length parameter relates the relative distance between the neighboring features. To estimate the upper bound of $\tilde{a}(i)$, consider $d_{i,i+1} > d_{i-1,i}$,

$$\tilde{a}(i) = \frac{\frac{3}{2}(d_{i-1,i} + d_{i,i+1})}{2 + \frac{d_{i,i+1}}{d_{i-1,i}}} = \frac{\frac{3}{2}(d_{i-1,i} + d_{i,i+1})}{\text{greater than } 3} = \frac{\frac{1}{2}(d_{i-1,i} + d_{i,i+1})}{\text{greater than } 1} \Rightarrow \tilde{a}(i) < \frac{1}{2}(d_{i-1,i} + d_{i,i+1})$$

Based on the length parameter and the distances $d_{i-1,i}$ and $d_{i,i+1}$ there can be two cases:

- If $|d_{i,i+1} - d_{i-1,i}| < \tilde{a}(i) \Rightarrow d_{i,i+1} \approx d_{i-1,i}$, this is defined as Case A
- Else if $|d_{i,i+1} - d_{i-1,i}| \geq \tilde{a}(i) \Rightarrow$ one of the two lengths, $d_{i,i+1}$ or $d_{i-1,i}$ is small as compared to other, this is defined as Case B

For $d_{i,i+1} = d_{i-1,i} \Rightarrow |d_{i,i+1} - d_{i-1,i}| = 0$ and $\tilde{a}(i) = d_{i-1,i}$, this indicates Case A. On the other hand if $d_{i,i+1} \gg d_{i-1,i} \Rightarrow |d_{i,i+1} - d_{i-1,i}| = d_{i,i+1}$. In Eqn.(4) using $\lim_{d_{i,i+1} \rightarrow 0} \frac{d_{i-1,i}}{d_{i,i+1}} \rightarrow 0$, $\tilde{a}(i)$ is obtained as $\frac{3}{2}d_{i-1,i}$ and since $d_{i,i+1} \gg d_{i-1,i}$, the set of values of $d_{i-1,i}$ and $d_{i,i+1}$ belong to Case B. Two examples considered validate the use of the above stated conditions.

It can be observed from Fig.4 that lengths $d_{i-2,i-1}$ and $d_{i-1,i}$ belong to case A while $d_{i-1,i}$ and $d_{i,i+1}$ belong to case B. It can also be seen that $d_{i,i+1}$ and $d_{i+1,i+2}$ belong to case A (here the two distances are almost equal in size but large as compared to any of the distances $d_{i-2,i-1}$ or $d_{i-1,i}$). Thus case A can be further divided into two sub cases; case A1 with almost equal but large lengths ($d_{i,i+1}$ and $d_{i+1,i+2}$) and case A2 with almost equal but small lengths ($d_{i-2,i-1}$ and $d_{i-1,i}$).

In order to identify the neighboring critical feature points, three consecutive feature points F_{I_m-1} , F_{I_m} and F_{I_m+1} are considered and the case to which they belong is determined. The procedure carried out based on the case is explained next.

Case A1: In this case since the lengths between the feature points is large, only F_{I_m} is considered as the critical feature point. Other feature point may or may not get included in the split segment depending on the minimum distance of point \mathbf{P} from the curve. If the distance D is large there is a possibility that the algorithm might include more feature points while computing the split length to avoid sudden change in the curve.

In finding the split ends, the splitting is performed at some distance away from the extreme critical features. To make sure that the features at the ends are also preserved, it is important to find this distance. This distance is identified by a length parameter ϵ defined as

$$\epsilon = d_{I_m-1, I_m} + d_{I_m, I_m+1} \quad (5)$$

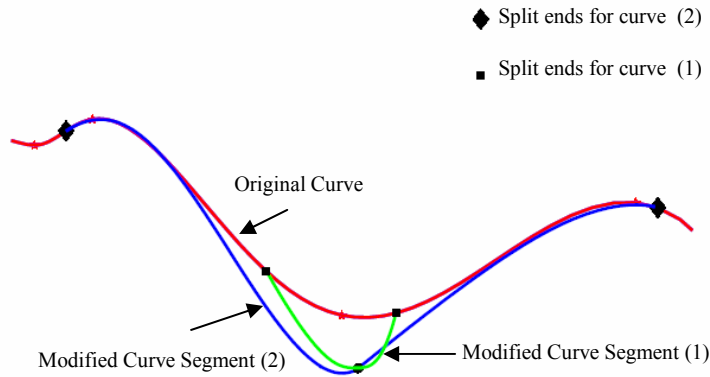


Fig. 5. Need to find Critical Feature points.

Case B: In this case one of the lengths d_{I_m-1, I_m} and d_{I_m, I_m+1} is small. Without loss of generality it can be assumed that $d_{I_m-1, I_m} < d_{I_m, I_m+1}$. Other critical feature points for this case are identified by searching for the features in the direction of the smaller feature length, d_{I_m-1, I_m} (Fig.6.). Through this, all the features with small distance $d_{i-1, i}$ in this direction ($F_{I_m-1} F_{I_m}$) are included as critical features. The inclusion terminates whenever a large distance is encountered. The last feature included is denoted as F_{end} .

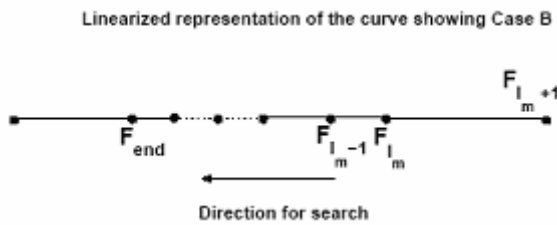


Fig. 6. Finding Critical Feature Points in Case B.

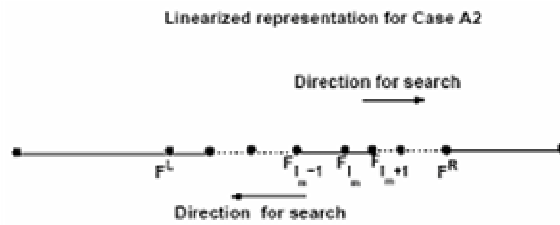


Fig. 7. Finding Critical Feature Points in Case A2.

Algorithm

1. Let $J=I_m$ and $Case(J)=case$ for $d_{J-1, J}$ and $d_{J, J+1}$
2. if $\{Case(J)=B\}$
 - F_J is included as a critical feature point
 - $J=J-1$
 - $Case(J)=case$ for $d_{J-1, J}$ and $d_{J, J+1}$
3. go to step 2
4. if the condition is no more satisfied $I_{end}=J$, F_{end} is the last feature included.
5. Terminate

For this case ϵ is defined as

$$\epsilon = \min((d_{I_{end}, I_{end}+1} + \dots + d_{I_m-1, I_m}), d_{I_m, I_m+1}) \tag{6}$$

Case A2: In this case since both the distances are small, the search is extended in both the directions until a large length is encountered (Fig.7.).

Algorithm

(First the features before F_{I_m} are identified)

1. Let $J=I_m-1$ and $Case(J)=case$ for $d_{J-1, J}$ and $d_{J, J+1}$

2. if {Case(J)=A}
 - F_j is included as a critical feature point
 - $J=J-1$
 - Case(J)=case for $d_{j-1,j}$ and d_{I_m-1,I_m}
3. go to step 2
4. if the condition is no more satisfied, $I_{endL}=J$, F^L is the last feature encountered in the left direction.
5. go to step 6

(Next the features after F_{I_m} are identified)

6. Let $J=I_m+1$ and Case(J)=case for $d_{j-1,j}$ and d_{I_m,I_m+1}
7. if {Case(J)=A}
 - F_j is included as a critical feature point
 - $J=J+1$
 - Case(J)=case for $d_{j,j+1}$ and d_{I_m,I_m+1}
8. go to step 7
9. if the condition is no more satisfied, $I_{endR}=J$, F^R is the last feature encountered in the right direction.
10. Terminate

For this case ϵ is defined as

$$\epsilon = \frac{d_{I_{endL}, I_{endL}+1} + \dots + d_{I_m-1, I_m} + d_{I_m, I_m+1} + \dots + d_{I_{endR}-1, I_{endR}}}{|I_{endR} - I_{endL}| + 1} \quad (7)$$

The identification of the critical feature points gives a rough estimate of the split length. The curve is split a small distance (determined by ϵ) away from the extreme feature points included as the critical feature. Define $\tilde{\epsilon} = \frac{\epsilon}{x}$, where $x (>2)$ is a factor that minimizes the value of split length. The optimal value of the split length is evaluated such that both the objectives of limited deviation and shape preserval are addressed. The shape of the curve is preserved by finding the critical feature points that would be affected by the shape change. Limited deviation is addressed by minimizing the value of $\tilde{\epsilon}$. If $\tilde{\epsilon} > d_m$ (Eqn.3.), the maximum value of x is obtained as $\max(x) = \epsilon/d_m$ where as if $\tilde{\epsilon} \leq d_m$ it is obtained as $\max(x) = 2 + \frac{\max(d_{I_m-1, I_m}, d_{I_m, I_m+1})}{\min(d_{I_m-1, I_m}, d_{I_m, I_m+1}, d_m)}$.

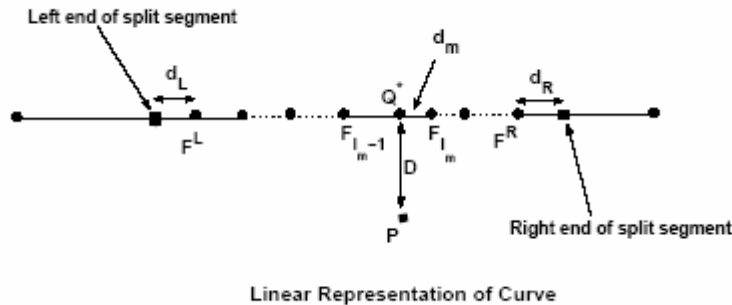


Fig. 8. Determination of Split Segment.

3.2 Determination of Split Ends

The splitting of the curve is carried out at some distance from the extreme critical feature points, evaluated at d_L and d_R from the left and the right extreme critical features (F^L, F^R) respectively (Fig. 8.). The procedure for obtaining the values of d_L and d_R is presented below.

Define $D_m = \sqrt{D^2 + d_m^2}$, $D_L = \text{length}(F^L, Q^*)$ and $D_R = \text{length}(F^R, Q^*)$.

Algorithm

- For i =left (L) or right (R)

- if $D_i < \tilde{\epsilon}$ then $d_i = (\tilde{\epsilon} + d_m)/2$
- else $d_i = (\tilde{\epsilon} + D_m)/2$

Determination of d_R and d_L completes the split end identification. Fig.9. shows the split ends for different positions of \mathbf{P} . Fig.9(a) to (c) indicates different cases based on the placement of \mathbf{P} and the determined split ends. In Fig.9(d) the curve is split in only two segments. This is observed because one of the split ends coincides with the curve end. It is thus indicative that placement of point \mathbf{P} is an important parameter to decide the curve segment to be modified. To split the curve, the parametric value of the point at which the curve has to be split is evaluated as u^* . The parameters corresponding to split ends are u_L^* and u_R^* respectively. Split segment is obtained when the above evaluated parameters (u_L^*, u_R^*) are inserted as knots of multiplicity p each in the knot vector U defined in Eqn.(1). [4].

4. RESULTS AND DISCUSSION

To illustrate the curve modification strategy into practice, MATLAB TM is employed to develop the prototype system. The prototype system was developed using personal computer with 512 MB RAM.

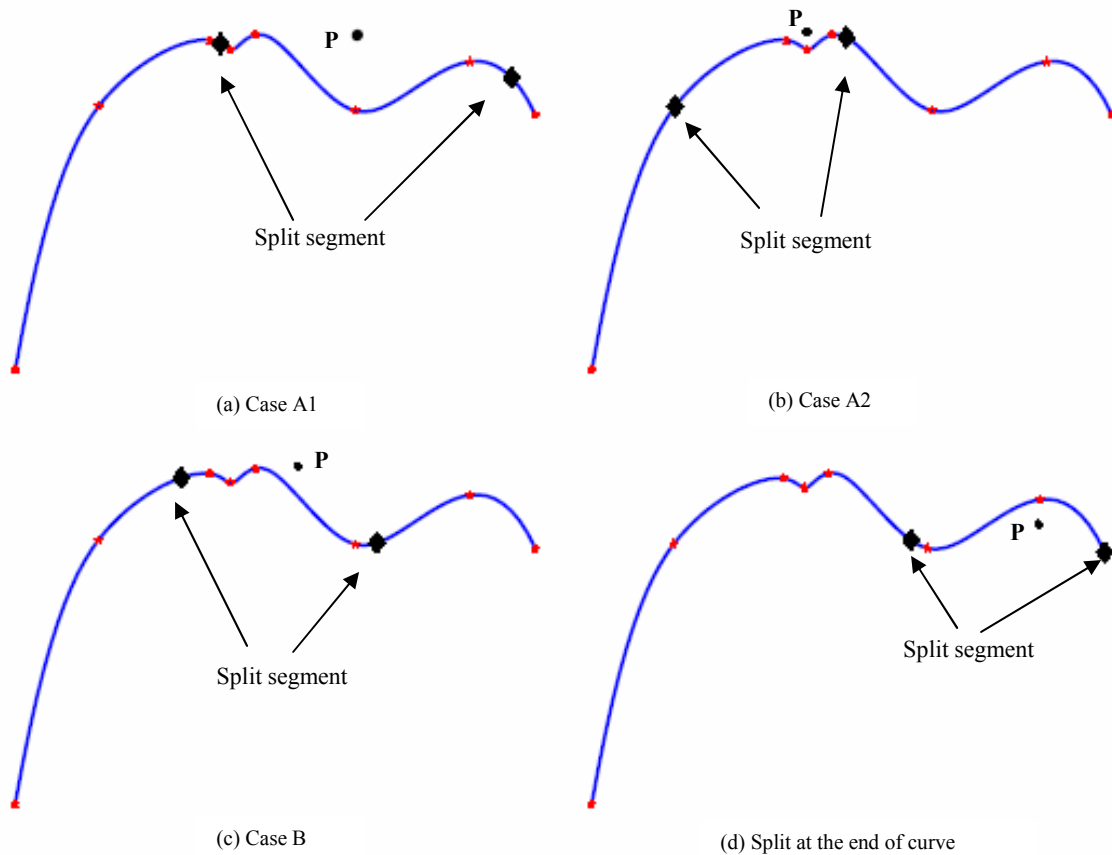


Fig. 9. Split Segment.

The original curve is split at the split ends, evaluated in section 3.2. Algorithm for constrained shape modification (section 2.2) is applied to the segment between the split ends. The splitting procedure explained in section 3, decides the number of segments after splitting. In any given curve and a given point \mathbf{P} there could be at most three segments. The segments in between the split ends determined is modified to achieve the desired objective. Application of constrained shape modification algorithm makes the curve in the split segment pass through the desired point. After the shape of the curve segment is modified, all the segments are combined to obtain the modified curve. This combining procedure involves removal of the knots inserted for splitting the curve (u_L^*, u_R^*) without changing the shape of any of the segments [4].

The use of splitting methodology explained restricts the deviation of the curve in a smaller region compared to the deviation which occurs without splitting. This is demonstrated in Fig.10. The curve in Fig.10(a) is obtained after splitting algorithm is applied with constrained shape modification whereas in Fig.10(b) the curve is not split. Fig.10(c)

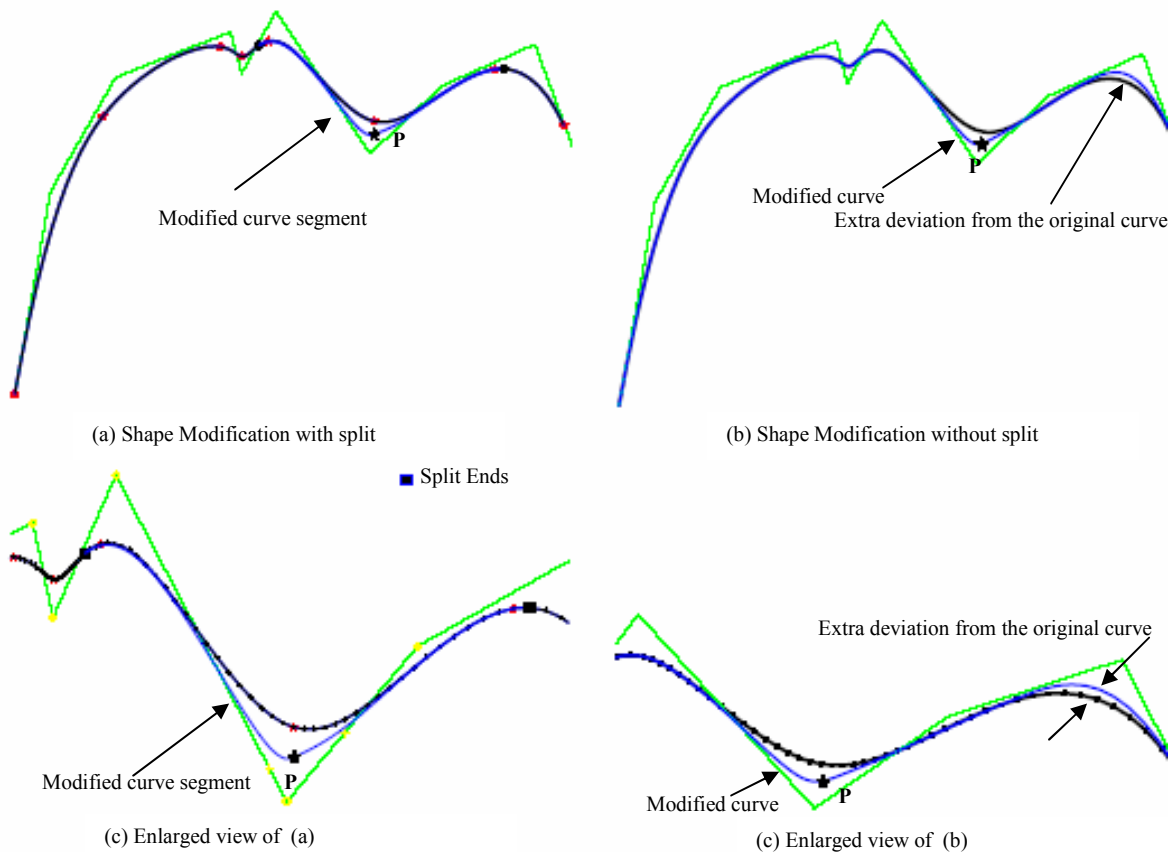


Fig. 10. Intrinsic Local Shape modification.

and Fig.10(d) show an enlarged view of the curves. It can be clearly seen from Fig.10(b) and Fig.10(d) that there is an extra deviation in the curve which is not observed when the curve is first split before shape modification (Fig.10(a),(c)). Thus splitting of the curve aids in, shape preservation and having limited deviation from the original curve without applying any optimization methods.

5. CONCLUSION

The present work aids in modifying the shape of the curve by making it pass through the desired point with the objectives of limited deviation and preserving its shape. Note that the shape modification algorithm discussed in section 2.2 is constrained for $\mathbf{P} \subseteq \mathbf{R}$, whereas the concept of splitting the curve and then modifying the split segment is applicable for any point \mathbf{P} . Though the constrained shape algorithm works for a restricted region \mathbf{R} , it is a single step non iterative scheme to make the curve pass through the desired point.

6. REFERENCES

- [1] Farin, G., *Curves and Surfaces for Computer Aided Geometric Design*, San Diego, Academic Press, 1997.
- [2] Boehm, W., Inserting new knots into B-spline curves, *Computer Aided Design*, Vol. 12, No. 4, 1980, pp 199-201.
- [3] Piegl, L., Modifying the shape of rational B-splines. part 1: Curves, *Computer Aided Design*, Vol. 21, No. 8, 1989, pp 509-518.

- [4] Piegl, L. and Tiller, W., *The NURBS Book*, Springer, Berlin, Germany, 1997.
- [5] Au, C. K. and Yuen, M. M. F., Unified approach to nurbs curve shape modification, *Computer Aided Design*, Vol. 27, No. 2, 1995, pp 85–93.
- [6] Fowler, B. and Bartels, R., Constrained based curve manipulation, *IEEE Computer Graphics Application*, Vol. 13, 1993, pp 43–49.
- [7] Chan, K. W., Zheng, J. M. and Gibson, I., A new approach for direct manipulation of free form curve, *Computer Graphics Forum*, Vol. 17, 1998, pp 327–334.
- [8] Juhasz, I. and Hoffmann, M., Constrained shape modification of cubic B-spline curves by mean of knots. *Computer Aided Design*, Vol. 36, 2004, pp 437–445.
- [9] Lyche, T. and Morken, K., The sensitivity of spline function to perturbation of the knots, *BIT*, Vol. 39, 1999, pp 305–322.
- [10] Piegl, L. and Tiller, W., Surface approximation to scanned data, *The Visual Computer*, Vol. 16, 2000, pp 386–395.
- [11] Cohen, F. and Wang, J., Modeling image curves using invariant 3-d object curve models- a path to 3-d recognition and shape estimation for image contours, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 1, 1995, pp 1–12.
- [12] Czuczor, S., Aszodi, B. and Kalos, L. S., NURBS fairing by knot vector optimization, *Journal of WSCG*, Vol. 12, No. 1.
- [13] Mamic, G. and Bennamoun, M., Automatic bayesian knot placement for spline fitting, *IEEE*, Vol. 1, 2001, pp 169–172.
- [14] Zheng, J., Sederberg, T. W. and Song, X., Knot intervals and multi degree splines, *Computer Aided Geometric Design*, Vol. 20, 2003, pp 455–468.
- [15] Wang, W., Yang, H. and Sun, J., Control point adjustment for b-spline curve approximation, *Computer Aided Geometric Design*, Vol. 36, 2004, pp 639–652.
- [16] Liu G. H., Wong Y. S., Zhang Y. F. and Loh H. T., Adaptive fairing of digitized data points with discrete curvature, *Computer Aided Design*, Vol. 34, 2002, pp 309–20.