Fitting B-splines to Scattered Data

New and Old Parameterization

Zexi Liu*, Fernand Cohen, Zhongchuan Zhang
Department of Electrical & Computer Engineering
Drexel University
3141 Chestnut Street, Philadelphia, PA 19104
{zl54, fscohen, zz57}@drexel.edu

***Abstract*—Parametric representation for curves is important in computer-aided geometric design, medical imaging, computer vision, computer graphics, shape matching, and face/object recognition. They are far better alternatives to free form representation, which are plagued with unboundedness and stability problems. This paper deals with the problem of fitting approximating B-spline curves with high compression ratios (control to data point ratios) to scattered data, where the data might be noisy or locally deformed, and where the curve sample points might be non-uniformly sampled across the curve. The approximating B-spline is robust to noise and local deformation. B-splines are bounded, continuous, invariant to affine and perspective transformations, and have local shape controllability. A parameterization based on curvature is proposed as a meaningful choice for the topological parameter of the B-spline that leads to a superior capturing of the curve shape and of preserving corner points. We compare several parameterization methods both theoretically and experimentally, and show that the curvature based parameterization is superior for noise free data, noisy data, local deformation, and non-uniform sampling. The fitting method is fast, and non-iterative. The fitting errors are, by and large, sub-resolution, i.e., they are below or at the resolution of the scattered data (defined here as the average distance between curve points). We also show the extension of the method to 3D surface fitting.**

Keywords: geometric features; corners; curvature; B-spline fitting

I. INTRODUCTION

A number of approaches have been proposed for curve modeling such as Fourier descriptors [1]; chain codes [2]; polygonal approximation [3]; parametric algebraic curves [4]; curvature invariant [5]; implicit polynomial functions [6]; elastic models [7]; and B-splines [8]. The B-spline stands as one of the most efficient curve representations, and possesses very attractive properties such as spatial uniqueness, boundedness and continuity, local shape controllability, and invariance to affine transformation. B-splines are parametric models that require a neighborhood relationship to be established between the data points prior to the construction of B-spline curve, as well as a topological (geometrical) meaningful assignment to the parameters. The topological parameter is usually taken as time or arc length, and the B-spline is looked upon as the boundary trace of a moving particle traveling in space. There have been different methods for assigning values for the topological parameters, each with its own different physical meaning, and with each leading to a different fitted B-spline. The easiest way of choosing a parameter is the uniform parameterization, which simply divides the parameter interval into equally populated segments. Clearly it cannot be expected that this simplistic method works well for arbitrary sets of data points, since the particular position of each point is not even considered. Then there is the so-called chord length (CL) method [9]. The parameter values are chosen in such a way that they reflect the length of the curve segments of the interplant between the data points. It assumes constant speed motion, which will be violated in the presence of local distortion or noise (non-uniform distributed noise) as it creates extra length of curve segments. It also suffers when the curve has areas of high and low shape variation that necessitates adopting different speeds while moving on the curve (non-uniform samples on the curve), with higher sampling rate in areas of high shape variation, and lower sampling rates in areas of low shape variation. The inverse chord length (ICL) method [10] is designed to overcome this drawback of CL. It makes parameter values inversely proportional to the chord length between data points, and hence contrary to the CL method, allocates more knot points (control points) to densely sampled and/or less noise contaminated curve segments. The centripetal parameterization [11] is a heuristic method which is very similar to the chord length method, but is inspired by a physical model. The parameter values are now based on the speed of a particle travels along the curve. Following the same idea that the parameter values should "slow down" in regions of high curvature, another purely heuristic method is introduced. It is called angular parameterization [12], which takes into account both the angle between the chords connecting the data points and the length of these chords. An upgrade to this method is the area based parameterization [13]. It is based on principles similar to those of the angular method, but is amiable to a reasonable geometric interpretation. In analogy to the angular method, two different contributing factors are considered for selecting the parameter values: the distance between two consecutive data points, and the inner angle between two chords. Finally, there is also work on fitting B-spline curves to point clouds that minimizes a fitting error term, defined by a curvature-based quadratic approximant of squared distances from data points to a fitting curve in an iterative optimization procedure [14].

In this paper, we develop a method that not only allows the

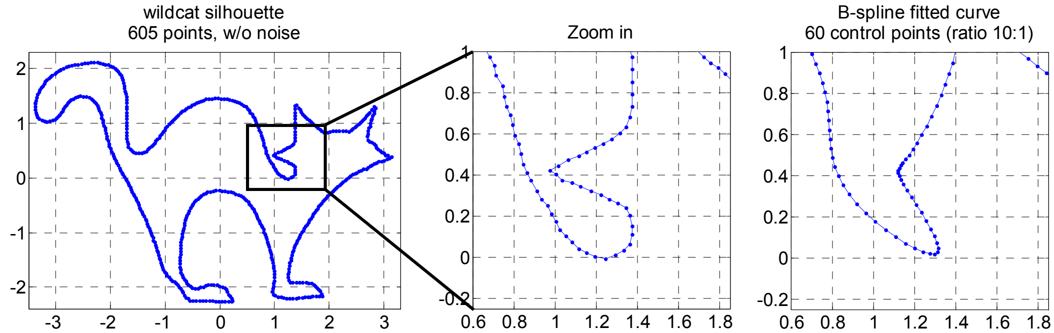


Figure 1. Corner lost during B-spline fitting

fitted B-spline curve to smooth out the noise of the raw measured curve but also allows it to "focus" on the high curvature areas. The motivation behind this is directly related to the corner point extraction/detection after B-spline fitting. The ultimate goal is to extract correct corner points on the measured curve with possible noises. To deal with noisy measured curves, an approximating B-spline filters out the noise by using a lesser number of control points than curve points. However, it may also smooth out the corners making them be impossible to detect or disappear altogether. Our proposed parameterization method makes use of the curvature information along the curve and finally leads to a feature preserved B-spline fitting.

This paper is organized as follows. Motivations for our feature preserved B-spline fitting is presented in Section II. The introduction of B-spline curves are given in Section III. The details of our parameterization method are given in Section IV. In Section V, the performances of all seven methods are investigated in a series of experiments. They will be compared both qualitatively by means of corner points detected and quantitatively by means of residual error analysis. Finally, conclusions and future directions are given in Section VI.

II. MOTIVATION

Corners usually represent critical information in describing object features that are essential for object identification. There are many applications that rely on the successful detection of corners, including curve and object alignment and registration, motion tracking, object recognition, and stereo matching. In the presence of noise resulting from bad edge localization and/or weak edge detection, a smoothing technique must be applied, which may lead to loss of information during smoothing. For example, a sharp corner will become a round corner after smoothing, or worse, it can disappear altogether after smoothing (see Fig. 1).

B-spline curves with a compression ratio of 10:1 fitted to the scattered points are shown in Fig. 1. Although adding more control points result into better preserved corners in the noiseless case, in the presence of noise however, it will result into a curve fitting that follows and accentuates the noise, and a worse corner and shape detection.

III. B-SPLINE CURVES

The B-spline curves are piecewise polynomial functions that provide local approximations to curves using a relatively small number of control points. A p^{th} degree (order $p+1$) B-spline is C^p continuous, i.e., is continuous and has its p derivatives continuous. A p^{th} degree B-spline with $n+1$ control points $C_0, C_1, \cdots, C_n$ consists of $n-p$ connected curve segments r_i, each of which is a linear combination of p polynomials of degree p in the parameter u, where u is normalized for each such segment between 0 and 1. The whole curve $r\ (u')$ $(0 \le u' \le n-2)$ consists of a concatenation of curve segments r_i that are continuously joint (the joining points are called the knot points). The knot points are the end points of the curve segments, i.e. they are points on the curve with $u' = 0, 1, \cdots, n-2$ (the B-spline knots). The control points are estimated by solving a least square problem [8]. For a p^{th} degree B-spline, the relationship between the control points and the knot points is as follows: every two consecutive knot points define a curve segment which is "controlled" by $p+1$ control points. Therefore, by assigning certain

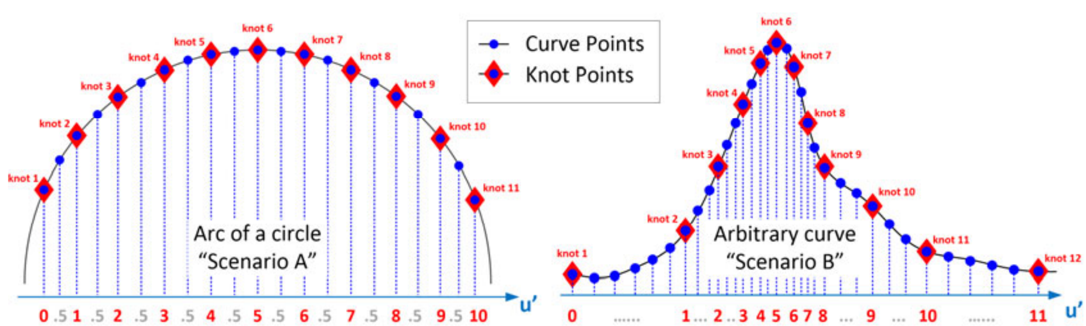


Figure 2. Knot points position

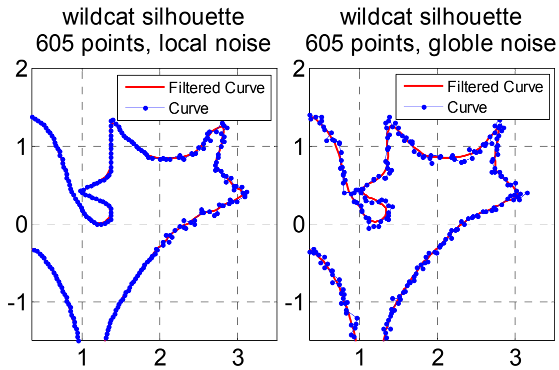


Figure 3. Experiment data

parameter values u' to each of the curve points, we could manipulate the position of knot points to change the distribution of control points which directly "control" the shape of the fitted curve. Fig. 2 illustrates this mechanism for two curves, the first of which basically exhibits constant curvature all over its parts, whereas the 2nd curve exhibits shape variations along its parts, by showing how the knots are distributed in these two cases, with the first having its knot uniformly distributed and positioned along the curve, whereas for the 2nd curve, they are non-uniformly positioned with more knots assigned to the high curvature parts. This basic principle is used throughout the seven methods for assigning u' values to the data points.

IV. CURVATURE BASED PARAMETERIZATION

The idea behind curvature based parameterization is to make the fitted B-spline curve "concentrate" on the high curvature areas, such as the corners with the effect of preserving them under an approximating B-spline fitting (otherwise sharp corners will become round corners). To achieve this goal, the curvature at each data point has to be pre-calculated. This can be done in two algorithms (both are tested in the experiments section):

1) Numerically calculate the curvature K_i at each point after applying a Gaussian filter that smooth the data.

$$K_i = \frac{\Delta x_i \cdot \Delta^2 y_i - \Delta^2 x_i \cdot \Delta y_i}{[(\Delta x_i)^2 + (\Delta y_i)^2]^{1.5}}, i = 1,2,\cdots,m \qquad (1)$$

Where $\Delta x_i = (x_{i+1} - x_{i-1})/2$, $\Delta y_i = (y_{i+1} - y_{i-1})/2$, $\Delta^2 x_i = (\Delta x_{i+1} - \Delta x_{i-1})/2$, *and* $\Delta^2 y_i = (\Delta y_{i+1} - \Delta y_{i-1})/2$.

2) Fitting a B-spline curve to the raw data using one of the methods outlined above, then compute the curvature associated at each data point from the fitted B-spline. This is readily available from the derivatives of the fitted B-spline.

The first algorithm will be referred to as superscript N and the second will be referred to as superscript F in the following text. With the curvature calculated at each curve point, the curvature based parameterization is expressed as follows:

$$u'_i = u'_{i-1} + \sqrt[\alpha]{K_{i-1}} \cdot \frac{u'_{max}}{\sum_{i=1}^{m-1} \sqrt[\alpha]{K_i}}, i = 2,3,\cdots,m \qquad (2)$$

where $u'_1 = 0$, and $u'_{max} = n - 2$ (for cubic splines), m is the total number of data points, $n + 1$ is the number of control points, and α is a positive number. The u' parameter assignment is in accordance with the difference between u'_i and u'_{i-1} being proportional to the curvature at data point $i - 1$. For segments with high curvature areas, more knot points will be allocated, and hence more control points will be assigned resulting in enhancing sharp corners.

The α parameter controls the degree of speed variation. The smaller the α is, the more control points are assigned to the higher curvature segments. The parameter α in (2) must be chosen based on the nature of the data set. If one or two really sharp corners exist on the curve, then a small α will result in over-fitting at those corners and under-fitting everywhere else. We experimentally found that $\alpha = 2$ gave the best results for the cases considered.

V. EXPERIMENTS

A series of experiments are designed to test all seven methods discussed above. The test objects are closed curves extracted from several 2D images (2000 by 2000 pixels) using a Canny edge detector, which yields a set of ordered data points. These were uniformly down-sampled to around 600 points (original over 6000 points) and referred as "raw curve" in the following sections.

Localized and global noise with Signal-to-Noise-Ratio of 30dB was added to the down-sampled curves (See Fig. 3). The SNR is calculated as

$$SNR_{dB} = 10 \cdot log_{10}\left(\frac{A_{signal}}{A_{noise}}\right) \qquad (3)$$

where A_{signal} is the mean square amplitude of the signal (curve), A_{noise} is the mean square amplitude of the noise which is generated using Matlab function *randn()*.

We have considered three cases: i) curves without noise, ii) curves with localized noise, and iii) curves with global noise. All seven different parameterization methods are tested for these 3 cases. We have used a 3rd degree B-spline and a compression ratio of 10:1 (1 control point for every 10 data points) in the experiments.

We report the results using two different error metrics: (i) the average residual error, which gives the average distance between B-spline fitted curve points and the raw curve points; and (ii) the maximum residual error which gives the maximum distance between B-spline fitted curve points and the raw curve points. We use a point-to-point distance map between the fitted B-spline curve and the raw curve, where for each pair of corresponding points the distance between them is computed. A point on the raw curve has its corresponding one on the fitted B-spline curve. Finally, both the average residual error and the maximum residual error are normalized so that they are expressed relative to the data resolution, which is taken here to be the average chord length of the points on the raw curve. The normalized errors are called relative errors and reflect the goodness of fitting better than the absolute errors. While the average residual errors associated with the different methods might all be sub-resolution or very close to each other, the

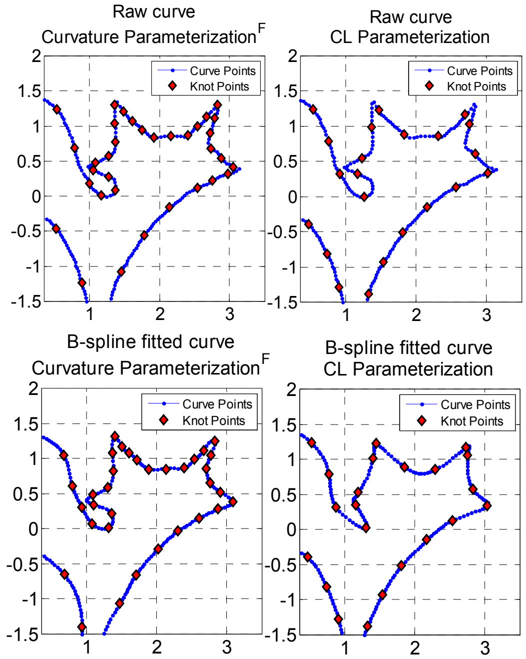


Figure 4. Knot points on the curves (without noise)

maximum residual errors usually occur on and around salient points such as sharp corners and hence would be a better measure of the goodness of fit. In all the figures we show the curvature based fitted curve as well as the worst fitted curve based on the maximal relative error.

A. B-spline fitting without noise

A B-spline is fitted to raw curve with 605 sample points on the Wildcat silhouette shown in Fig. 1 (most left) using the various parameterization methods outlined above. In Fig. 4, the shape and the corners are well preserved using the curvature based parameterization but not so well using the uniform, CL, ICL, and etc. Table I shows the relative residual error of the B-spline fitted curve using the various parameterization methods at a compression ratio of 10:1. As we can see from the table, the average error doesn't fully reflect the goodness of fit. The errors are about the same as or below resolution. The maximal relative error, however, gives a better ranking of the various methods, with the best fit achieved using our new curvature method while the other methods produce errors two or three times larger than the resolution. This is also clearly reflected in Fig. 8, where the curve's shape with its corners is far better captured using the curvature methods. It is also interesting to note that for the curvature method the maximum errors are still sub-resolution (i.e., they are at 87.60% of the curve resolution).

TABLE I. ERROR EVALUATION – NO NOISE

	Average Error	Maximum Error
Uniform	102.27%	288.07%
CL	100.38%	304.39%
ICL	101.27%	283.03%
Centripetal	103.02%	296.89%
Angular	77.58%	287.95%
Area	81.80%	266.77%
Curvature^F	25.19%	87.60%
Curvature^N	21.63%	102.49%

From now on, the figures will always show the lowest and the highest relative maximal error among all seven methods. In Fig. 4, CL method has the worst performance (highest relative maximum error), Curvature method is the best. Area method is the second best (See Table I).

B. B-spline fitting with localized noise

We repeat the previous experiment but now for the case when part of the curve is contaminated with white Gaussian noise of SNR 30dB. The noise was concentrated around the cat's head. The B-spline was fitted using all of the u' assignment methods, and the errors are measured relative to the raw curve (the noise free curve) and not the noisy curve (See Fig. 5). As seen from Table II, the curvature based method yields the best results with sub-resolution maximal relative error (92.47%), followed by the Centripetal (219.41%) with the worst fit being the area method (377.29%). It is interesting to see that the curvature method was able to filter out the noise while preserving the shape and corners.

C. B-spline fitting with global noise

In this scenario, the entire curve is buried in the noise with SNR of 30dB. The results are shown in Fig. 6 and Table III. The curvature based method still outperforms all other methods with maximal relative errors ranging between (112.04%-118.36%), with all others methods yielding far bigger values. In spite of the noise, the method still preserves the original shapes and the corners far better than the other methods.

D. Iterative B-spline Parameter Estimation

This section uses the different u_i' assignment methods as the initial guess for further optimization. The goal here is to show whether or not further optimization leads to the same optimal solution for all the methods mentioned in this paper as well as to determine if further optimization is warranted to start

TABLE II. ERROR EVALUATION – LOCALIZED NOISE

	Average Error	Maximum Error
Uniform	104.89%	287.36%
CL	93.17%	313.22%
ICL	110.94%	298.67%
Centripetal	72.45%	219.41%
Angular	89.51%	292.90%
Area	111.07%	377.29%
Curvature^F	31.01%	92.47%
Curvature^N	33.93%	101.04%

TABLE III. ERROR EVALUATION – GLOBAL NOISE

	Average Error	Maximum Error
Uniform	104.40%	282.82%
CL	100.68%	301.26%
ICL	103.05%	257.58%
Centripetal	103.85%	304.80%
Angular	114.26%	303.17%
Area	90.34%	350.66%
Curvature^F	45.16%	112.04%
Curvature^N	48.36%	118.36%

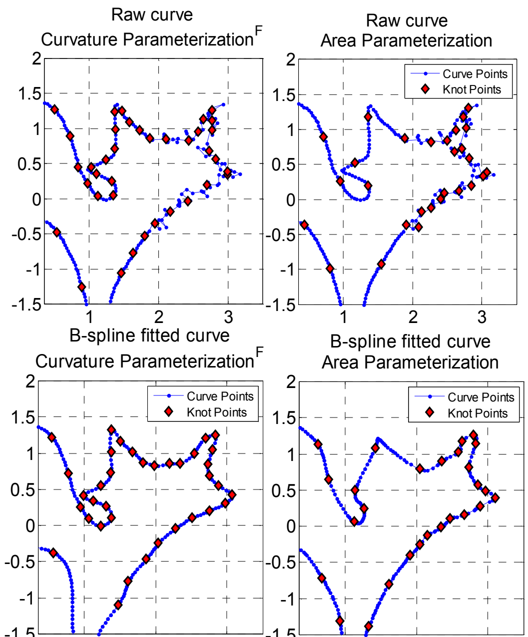


Figure 5. Knot points on the curves (localized noise)

with.

We adopt an iterative procedure similar in spirit to the EM algorithm in maximum likelihood estimation [15]. The steps in this iterative minimization process are as follows:

1) *Initialization: Choose initial values of* u_i', *and denote them as* $u_i'(old)$.
2) *Compute the control points* $C_i(old)$ *based on the assigned* $u_i'(old)$ *values, and compute the mean residual errors* $e(old)$.
3) *Update the values of* $u_i'(old)$ *by minimizing mean residual error* e *with the control points fixed to* $C_i(old)$. *Based on* $u_i'(new)$, *find* $C_i(new)$ *and compute* $e(new)$.
4) *If* $e(new) < e(old)$ *and* $|e(new) - e(old)| > threshold$, *then set* $C_i(new)$ *to* $C_i(old)$, $e(new)$ *to* $e(old)$, *and go back to 3; otherwise go to 5.*
5) *Stop. The control points are* $C_i(old)$ *in step 3.*

This iterative minimization is guaranteed to converge to at least a local minimum of e. The initial assignment for u_i' values is made according to the methods we introduced in section V. The update for the values of u_i' in step 3 for a fixed set of control points is achieved according to the following. Let $u_i'(old)$ be the value of u_i' before the update. First, we confine the search region to the interval $I = [Integer\{u_i'(old)\} - 1, Integer\{u_i'(old)\} + 1]$. This interval corresponds to the three B-spline curve segments confined by the knots $k-1$, k, $k+1$, and $k+2$, where $k = integer\{u_i'(old)\}$. The best value $u_i'(new)$ for u_i' that minimizes e confined to that interval is found using a golden section search [16]. The update for the next parametric value u_{i+1}' is obtained in a similar fashion, except that it is not allowed to assume a value smaller than $u_i'(new)$, i.e., the search imposes the order constraint

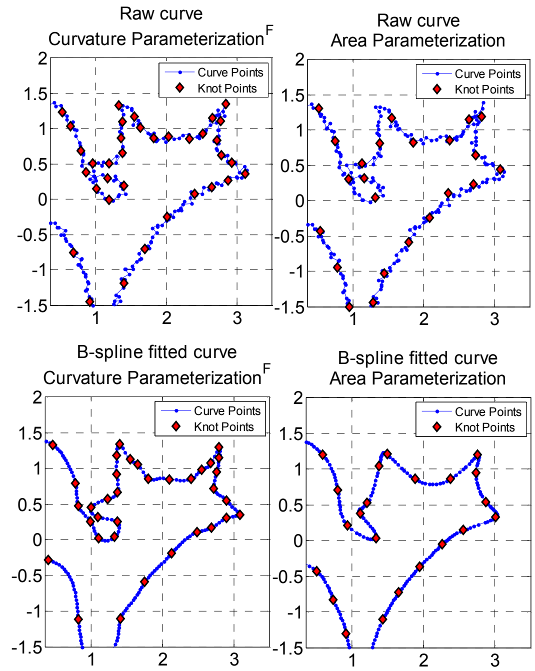


Figure 6. Knot points on the curves (global noise)

$u_1' < u_2' < \cdots < u_i'$.

We performed an experiment on the wildcat data shown in Fig. 1. The test results are shown in Fig. 7. All methods improved on the error metric. However, they didn't all converge to the same value. As we can see from Fig. 7 below (x-axis is the number of iterations; y-axis is the normalized B-spline fitting errors), our curvature method had the lowest convergence error value when compared to all the other methods convergence values. In addition, the initial value for our method is far below all the error values that the other methods converge to. This means that where we start matters and that the initial value for our curvature based method is an improvement over all the other methods even when we allow for further optimization for these methods. Furthermore, we can see that the curvature-based method initial guess had a relative maximal sub-resolution error value (87.60% as shown

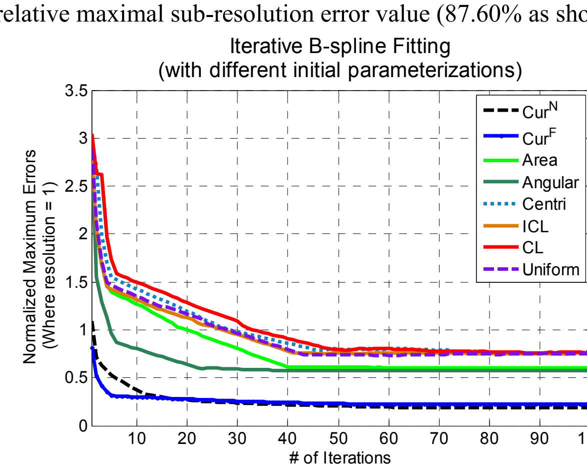


Figure 7: Maximum Error evaluation of iterative B-spline fitting

in Table I and 92.47% in Table II for the noisy data) that was close to the optimum convergence values associated with all other methods when we allowed for further optimization. We can also make the argument that since the initial relative maximal error is sub-resolution that further optimization for the curvature based method is not really warranted. This is, however, not true for all other methods, whose maximal relative error values are not sub-resolution.

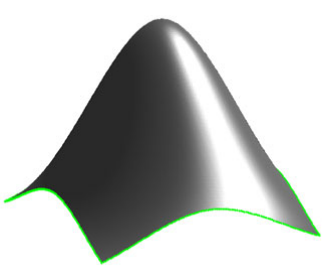
E. 3D Surface Fitting

The curvature based parameterization method can be easily extended to 3D domain. In 3D B-spline surface fitting, the parameter plane (the u-v plane) is partitioned in to Δu and Δv bins, where each bin is a 3D curve. The data points that fall in each bin are ordered as 3D curve points to which the curvature based assignment introduced earlier is the paper is directly applied. This curvature based assignment results into having more knot (control) points on the high curvature area (See Fig. 8) and less on the low curvature areas.

VI. CONCLUSIONS

We presented a curvature based assigning for the topological parameter of the B-spline and showed how it affects the approximating B-spline fit. We showed that this assignment is robust to local and global noise and achieves very high compression ratio (ratio of data to control points) by allocating fewer control or knot points to regions on the curve with small shape variation (flat area) while allocating more control points to regions of high shape variation (sharp corners). Under the same compression ratio, it gives the best results when compared to all other existing methods: uniform, CL, ICL, Centripetal, Angular, and Area parameterization. The errors obtained using this new curvature based assignment method is, by and large, sub-resolution, i.e., they are below or at the resolution of the scattered data (average distance between curve points). Furthermore, this method performs very well for feature point extraction (for object recognition and registration). Using the same extraction algorithm [17] is, the fitted curve using curvature parameterization gives the most reliable number of corner points (See Fig. 9), which are essential in image registration, as missing feature points or incorrect feature points may lead to a misalignment.

Surface with high curvature in the central area



Control polygon on the synthesized surface

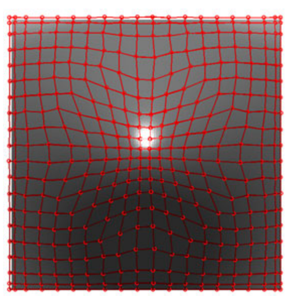


Figure 8. An example for 3D B-spline fitting

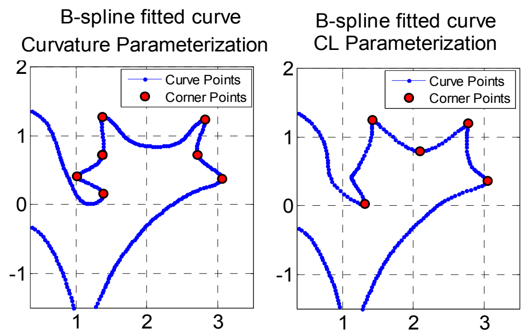


Figure 9. An application in feature points extraction

REFERENCES

[1] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, pp. 388-397, May 1986.

[2] J. A. Saghri and H. Freeman, "Analysis of the precision of generalized chain codes for the representation of planar curves," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 3, pp. 533-539, Sept. 1981.

[3] T. Pavlidis and F. Ali, "Computer recognition of handwritten numerals by polygonal approximations," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-6, pp. 610-614, Nov. 1975.

[4] J. Ponce and D. J. Kriegman, "On recognizing and positioning curved 3d objects from image contours," in Proc. IEEE Workshop Interpret 3-D Scenes, Nov. 1989.

[5] P. J. Besl and R. C. Jain, "3-D object recognition," ACM Computing Surveys (CSUR), vol. 17, Mar. 1985.

[6] D. Forsyth, J. Mundy, A. Zisserinan, and C. Brown, "Projective invariant representation using implicit algebraic curves," in Proc. Europ. Conf. Comput. Vision, 1990.

[7] S. H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn, "A Novel Representation for Riemannian Analysis of Elastic Curves in Rn," presented at the CVPR, Minneapolis, Minnesota, USA, 2007.

[8] L. A. Piegl and W. Tiller, The NURBS book, 2nd ed. Berlin ; New York: Springer, 1997.

[9] F. S. Cohen and J. Y. Wang, "Modeling image curves using invariant 3-D object curve models-A path to 3-D recognition and shape estimation from image contours," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, pp. 1-12, Jan. 1994.

[10] Z. H. Huang and F. S. Cohen, "Affine-invariant B-spline moments for curve matching," IEEE Transactions on Image Processing, vol. 5, pp. 1473-1480, Oct 1996.

[11] E. T. Y. Lee, "Choosing nodes in parametric curve interpolation," Computer-Aided Design, vol. 21, pp. 363-370, July/August 1989.

[12] T. A. Foley and G. M. Nielson, Knot selection for parametric spline interpolation, 1989.

[13] W. Heidrich, "Spline Extensions for the MAPLE Plot System," Master, Department of Computer Science, University of Waterloo, 1995.

[14] W. Wang, H. Pottmann, and Y. Liu, "Insert Fitting B-Spline Curves to Point Clouds by Curvature-Based Squared Distance Minimization," ACM Transactions on Graphics, vol. 25, pp. 214-238, 2006.

[15] F. S. Cohen, Z. Huang, and Z. Yang, "Invariant Matching and Identification of Curves Using B-Splines Curve Representation," IEEE Transactions on Image Processing, vol. 4, Jan 1995.

[16] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes in C. UK: Cambridge University Press, 1988.

[17] X. C. He and N. H. C. Yung, "Corner detector based on global and local curvature properties," Optical Engineering, vol. 47, pp. 1-12, 2008.