

UNCLASSIFIED

Defense Technical Information Center
Compilation Part Notice

ADP012052

TITLE: Adaptive Parameterization and Approximation for CAD Data Reduction

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: International Conference on Curves and Surfaces [4th], Saint-Malo, France, 1-7 July 1999. Proceedings, Volume 1. Curve and Surface Design

To order the complete compilation report, use: ADA399461

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:

ADP012010 thru ADP012054

UNCLASSIFIED

Adaptive Parameterization and Approximation for CAD Data Reduction

G. Wahu, J. M. Brun, and A. Bouras

Abstract. Data reduction is frequently needed in curve and surface model conversion, primarily for CAD data exchanges, but also to ease the designer's work. Such data reduction corresponds globally to a number of pole reductions. A previous analysis has shown that there are two main criteria in this domain: the parameterization and the extremity conditions. The possibility of defining an optimum optimum solution for the approximations used in data reduction is noted. The relation between this optimum optimum and the existence of an optimal parameterization leads to a new approach for curve and surface approximation. This approach has the advantage of modifying the parameter settings in a transparent way, while matching easily the extremity conditions. Finally, the extension to surface data reduction of this scheme is presented.

§1. Introduction

Designers have to create shapes in highly constrained environments. Curves and surfaces have to join precisely under tangency and curvature conditions; they have to meet some points as precisely as possible while behaving "nicely" between these points. All these conditions result into curves and surfaces either over-segmented or of dangerously high degree, and eventually both. Such curves and surfaces that are defined by a number of poles larger than necessary, induce severe problems in further use. Data exchange with other systems can be impossible if the degree exceeds what is allowed in the receiving system, or untractable if the number of curves or surfaces generated is too large. Data quality can be very poor. Curves and surfaces which are defined by an extremely large number of poles have difficulties in behaving "nicely". One knows also that the parameterization of curves and surfaces is crucial when defining them by discrete points [7,9,10,12]. Strangely, it is generally considered that CAD data reduction must be done with the original parameterization.

One considers also that the quality of a data reduction is directly related to the distance between points of a same parameter value, doing so

one mixes distances perpendicular to the original curve with tangential distances corresponding to different parametrizations. One must remark also that some approaches cannot meet essential conditions such as end conditions. We present here some results of our search for the optimal parametrization which minimizes the amount of data needed to fit curves and surfaces to a given precision.

§2. Previous Approaches

2.1 Degree reduction of Bézier curves

This problem was addressed early and is considered as the inverse problem to degree elevation [3,4,5,6,7,11,14]. Degree elevation is obtained step by step by a de Casteljau process of poles creation, which can be done either from poles π_0 to π_n or π_n to π_0 which produces two different results depending on the way it is processed. The idea of a blend between these two elementary processes, with blending coefficients depending on the rank in each elementary process, is a natural one. It was an important insight in this process that enabled Eck [5,6] to find the optimal blending and to prove that it is the best componentwise approximation. However, this optimal solution, aside from its componentwise limitation, assumes that the curve parametrization stays unchanged. Improvements over the Eck's solution can be expected from a global optimization and parametrization modifications.

2.2 Reduction in the number of poles for B-splines

The reduction of the number of poles is a crucial problem when converting from B-Splines to the Bézier form. B-Splines can be decomposed into Bézier curves or patches of low degree (typically 3 with C^2 continuity). Doing so produces huge sets of low degree Bézier curves or patches, which are unmanageable on most Bézier based CAD systems.

2.3 Classical parametrization schemes

Classical parametrization schemes were presented by Farin [7]. His conclusions are that it is good practice to test chordal parametrization first, then Lee's centripetal scheme and ultimately Foley's tangent variations. Such parametrizations come from kinematic analogies where one travels on the curve at constant speed or slows down on curves depending on centrifugal forces or the speed of turning the steering wheel. These schemes, while grounded in common sense, were probably found too empiric by Hoscheck [10], who proposed a scheme relating parameter modifications to tangential errors. In this scheme, for a given parametrization, a least square minimization produces a curve minimizing both normal and tangential errors. Iterative modifications of the parametrization using, at each step, the parametrization produced by the preceding least square modifications would be extremely computer intensive. Hoscheck uses instead a projection of the errors on the curve tangents for each parameter value, in order to optimize the parametrization. However, the

process remains computer intensive since it still has to loop on least square minimizations. Moreover the process can have convergency difficulties that can stop the process far from it's goal. This goal is to reach a parametrization producing normal errors, called "intrinsic parametrization" by Hoscheck. The optimal parametrization sought here can be defined for the approximation of curves or surfaces known everywhere, or nearly everywhere. It is the parametrization for which the maximum distances between each point on a curve and the approximated curve is minimal, which implies that these distances are perpendicular to the curve.

§3. Proposed Approach

3.1 Early approaches to optimal parametrization

In his thesis, P. Bézier [1] considers implicitly that a parametrization proportional to curvilinear abscissa is optimal, and that chordal parametrization of sparse sets of points is an approximation that can be slightly improved taking circles passing through points taken 3 by 3. At the same time (1975) J. M. Brun, in a non published study, took the Bézier curve defined by the sparse set of points, taken as poles of the curve, and used their projections on this curve as parameters for the points. He found that improvements over this parametrization are possible when an iterative modification of the curve is made using normal errors to move poles and tangential errors to modify the parametrization. The computing power available by that time imposed to stop the process after a small number of iterations, and the scarce set of points prevented to define what can be an optimal parametrization, since the definition of a distance between curves was not possible.

3.2 Our search of an optimal parametrization

Following similar goals as Hoscheck [10], we have tried to define intrinsic parametrization of curves, related only to shape characteristics. The first characteristic of a curve is it's length, corresponding to the curvilinear abscissa s as an intrinsic parametrization of the curve. Then, any other parametrization can be defined as a parametrization law: $t = f(s)$. We found that a mathematically sound strategy can be to approximate the optimal law $t = f(s)$ by a power series $t = s(a + b * s + c * s^2 + \dots)$ [2]. The coefficients of this series can be derived from geometric extremity conditions: end points produce $t = s$ (called a linear law), end tangents produce $t = s(a + b * s)$ (called a parabolic law) and adding curvatures produce $t = s(a + b * s + c * s^2)$ (called a cubic law). Depending on the curve shapes, the law to use needs more or less completion, and the results are convincing up to curve shapes with one inflexion. For curve shapes of higher complexity, higher degree series are needed, and the cubic scheme was found complex enough to avoid going further. In such cases, a segmentation of the curve allowing a piecewise approximation of the $t = f(s)$ law was envisioned.

3.3 Are designer's skills inherently better than mathematics?

While experimenting with our mathematical approaches, either improving the degree of $t = f(s)$ or taking care of curvature extrema, it was found that an experienced designer was always able to improve the result [13]. He just moves poles interactively. The parametrization of the designer's curve produces a curve significantly closer to the target curve than any mathematical approach. It was thus considered of interest to analyze the designer's actions, and to reproduce them by appropriate heuristics.

3.4 The designer's algorithm

The experienced designer knows the influence of a pole displacement and "integrates" the errors on the curve (perpendicular to the target curve) as "demands" to modify each pole with "weights" implicitly given by the coefficients $B_i^n(t_j)$, heavily influenced by his experience and adjusted by the feedback of the curve response to his modifications. In this process of curve adjustment by displacing poles, he simultaneously modifies the curve parametrization and the curve shape until he obtains a global optimum for which any pole displacement would increase normal errors at some places more than it would reduce it at others. The experienced designer knows pretty well that a pole displacement has no influence on the normal errors at places parallel to that displacement. In doing so, he has the ability to reduce errors by parametrization improvements, even though he is generally unaware of it. This analysis of the rationale behind the designer's heuristics produces an algorithm that can be called the designer's algorithm:

Extract the sample points P_j from the original curve:

The number of points P_j has a direct influence on the performance of the process. This sample of points can be refined to improve the precision of the approximation and to obtain better convergence.

Construct an initialization curve.

Iterate while the process converges until the desired precision is reached:

- 1) *For each point P_j , seek for the point $C(t_j)$ of the approximation curve whose normal passes through P_j ,*
- 2) *Check the precision of the approximation,*
- 3) *Find the maximum errors,*
- 4) *For each pole, calculate and apply a "displacement demand",*
- 5) *Study the convergence of the process.*

The "displacement demand" on a pole is that $\delta\pi_i = \sum_j (P_j - C(t_j))B_i^n(t_j)$, see Fig. 1.

3.5 The math behind the 'designer's algorithm'

When the designer's algorithm reaches its goal, all the errors are normal to the approximating curve, and the parametrization of the given curve is given

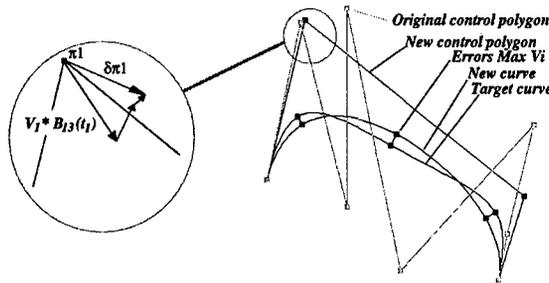


Fig. 1. The designer's algorithm.

by the approximation curve $C(t)$. With such a parametrization, the least square approach minimizes the error function:

$$E = \sum_{j=1}^m (P_j - C(t_j))^2,$$

where P_j are again the sampling points, and the approximation is

$$C(t_j) = \sum_{i=0}^n \pi_i B_i^n(t_j).$$

The minimizing conditions $\frac{\delta E}{\delta \pi_i} = 0$ leads to

$$\sum_{j=1}^m (P_j - \sum_{k=0}^n \pi_k B_k^n(t_j)) B_i^n(t_j) = 0; i \in [0, n].$$

For G^0 conditions, $\pi_0 = P_1$ and $\pi_n = P_m$ are the curve's extremities. For G^1 conditions, π_1 and π_{n-1} are constrained to lay on the extremity tangents. For convergence, we can identify the following least square conditions on the designer's algorithm:

$$\sum_{j=1}^m (P_j - C(t_j)) B_i^n(t_j) = 0; i \in [0, n].$$

After convergence, the designer's algorithm produces

- a parametrization where errors are normal to the curve,
- the least square solution for this parametrization.

The existence and uniqueness of this solution relies on the convergence of the process and the uniqueness of the parametrization. Aside from proofs of the convergence that mathematicians may provide, we can say that the

convergence of the process is typically slow but regular. However, single iterations are much faster than the mean square resolutions used by Hoscheck. Globally, the performance can be compared favorably with Hoscheck's intrinsic parametrization, which has the same goal of normal errors but a convergence that can be irregular.

Using a mastering of the process convergence [8], we were able to improve considerably the convergence rate of a regular convergence such as for the designer's algorithm. Depending on the initializing curve, the process can fall in a "convergence trap". To get out such traps, one can envisage using simulated annealing techniques. However, we have observed that our convergence mastering techniques combine jumping over traps to process acceleration. The uniqueness of the parametrization would be ensured by errors normal to the given curve, but it can be questionable for errors normal to the approximating curve. Aside from mathematical proofs, we remark that when the approximation is good, both normals are equivalent and so are the parametrizations. Moreover, if one minimizes the maximum errors only, as with the Tchebychev minimax approach, the normals at these points are common to both curves. This corresponds to a real designer's actions who takes care of maximum errors only, so the designer's algorithm has to use a "displacement demand":

$$\sum_{k=1}^l \vec{\delta}_i k, \quad i \in [0, n],$$

$$\vec{\delta}_i k = \text{maxk}((P_j - C(t_j))B_i^n(t_j)).$$

The *maxk* function produces the maximum value on the interval k between two crossings of P_j and $C(t_j)$. This displacement demand corresponds to a curve's distance which is a mix of the least square distance and the minimax of Tchebychev. When using this distance, one speeds up the algorithm and produces a better accuracy since the result is closer to an equioscillating approximation. An extension of the 'designer's algorithm' to B-Splines or NURBS would be straight forward: one has just to replace the $B_i^n(t)$ weighting function by corresponding ones. However, node sequence modifications are not done implicitly by such extensions, and an effective designer's algorithm must include node sequence definition. The extension to surfaces is also straight forward, but it has the additional advantage of working where designers may have trouble moving poles interactively. This extension is the following:

- 1) Reduce boundary curves complexity and adjust degree on opposite curves.
- 2) Use a Coons bilinear interpolant of the boundary curves as starting point.
- 3) Iterate on poles' displacements like for curves.

Pole displacements are modified by the use of errors normal to the approximating surface $S(u, v)$ and weighting functions $B_i^n(u) B_j^m(v)$ in place of $C(t)$ and $B_i^n(t)$. Then

$$\vec{\delta} \pi_{ij} = \sum_{i=0}^n \sum_{j=0}^m (P_{kl} - S(u_k, v_l)) B_i^n(u_k) B_j^m(v_l); i \in [0, n], j \in [0, m],$$

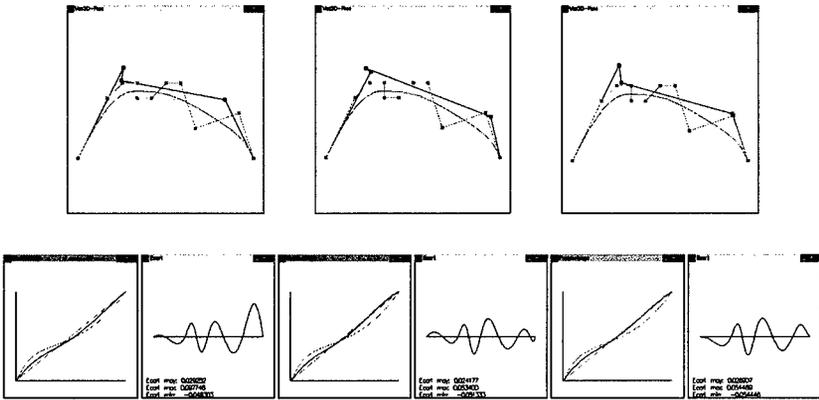


Fig. 2. (a) G^0 Hoschek, (b)–(c): G^0 and C^1 designer's algorithm.

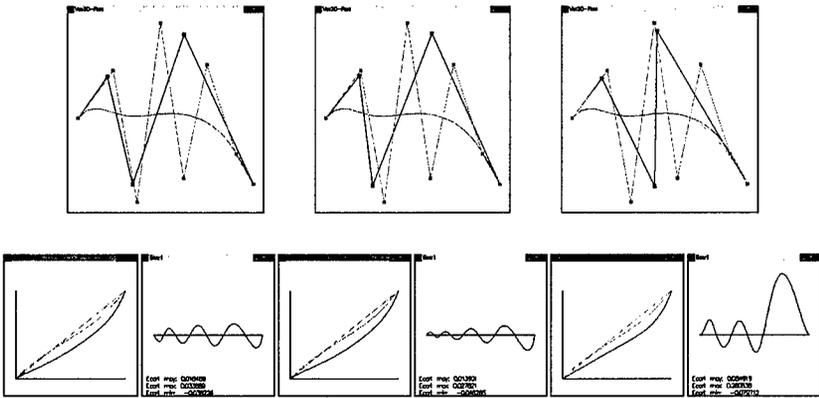


Fig. 3. (a) G^0 Hoschek, (b)–(c): G^0 and C^1 designer's algorithm.

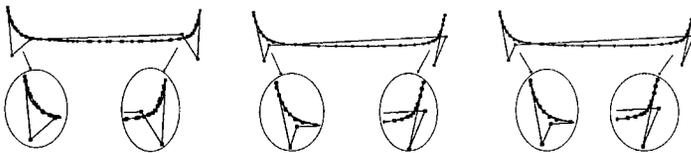


Fig. 4. (a) G^0 Hoschek, (b)–(c): G^0 and C^1 designer's algorithm.

where $S(u, v) = \sum_{i=0}^n \sum_{j=0}^m \pi_{ij} B_i^n(u) B_j^m(v)$. The 'designer's algorithm' on surfaces has the same behavior as for curves, but the computing time is obviously higher and the convergence mastering much more important. Replacing the least square distance by the local maxima distance would speed up the process quite efficiently like for curves.

§4. Results

4.1 Bézier and B-spline curves

The first test uses a degree-10 Bézier curve, proposed by Bogacki [3]. The first approximations are calculated by using the Hoschek's G^0 solution (Fig.2a). The next curves are the G^0 (Fig.2b) and G^1 (Fig.2c) approximations obtained after application of the proposed 'designer's algorithm'. The precision are respectively: 0.029, 0.024, 0.027. Note our process adapts nicely to the G^1 condition.

The second test curve (Fig.3) is a 'real' Bézier curve, found in CAD data transfer. Designers feel comfortable in reducing from degree 7 to 4 at the cost of the cancellation of a small inflexion close to the right extremity. Here the smoothed inflexion degrades the designer's algorithm curve when a G^1 condition is needed. These curves precision are respectively 0.018 for Hoschek's G^0 curves, and 0.013, 0.084 for the G^0 and G^1 designer's algorithm curves.

The last test curve (Fig.4) is also a 'real' curve. It is a B-Spline curve defined by 28 poles with a parametrization law more or less linear. This curve has a somewhat chaotic curvature repartition that can be smoothed vigorously with a degree 5 Bézier curve. The designer's algorithm process again gives much better results than Hoschek's solution with the G^0 condition. The precision obtained is 0.00138 for Hoschek's G^0 curve, and 0.00095 and 0.00097 for our G^0 and G^1 curves, respectively. Since we share with Hoschek the principal of searching a parametrization producing normal errors, our better result is due to better and more regular convergence.

4.2 Surfaces

Surfaces are computationally much more expensive than curves, but the possible data reduction is also much higher. Consider for example a B-spline surface defined by a network of $23 * 23$ poles (Fig. 5a).

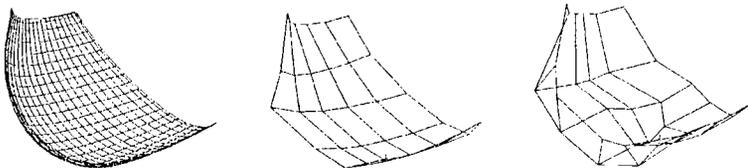


Fig. 5. Control polygons: Original (a) Coons/Bézier (b) 'Designer's Algorithm' (c).

Since it is a $3 * 3$ degree B-spline surface, the number of poles would be $16 * 22 * 22$, if converted to a set of $3 * 3$ degree Bézier patches. It is possible to convert it to a $5 * 5$ degree Bézier surface of 36 poles only. We use the degree-5*5 Coons/Bézier surface (Fig.5b) as initial surface. Figure (Fig.6a) represents the evolution of the error along the surface. At this stage of the process, the approximation's relative error is 0.0475. Then the designer's algorithm is

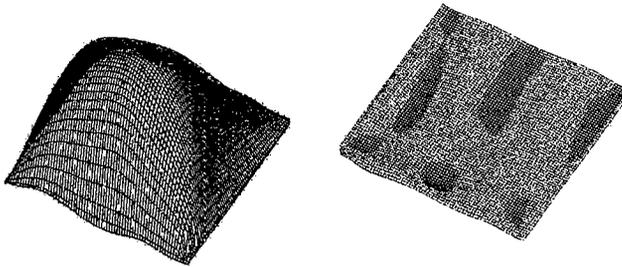


Fig. 6. Errors: Coons/Bézier (a) 'Designer's Algorithm' (b).

applied to the surface (Fig. 5b), and one obtains the surface (Fig. 5c) whose relative error is only 0.00126.

§5. Conclusion

Reducing the number of poles is an important problem for converting curve and surface models, and is needed in CAD data exchanges. Reducing step by step the degree of a Bézier curve shows that the original parametrization is less and less optimal. More generally, the approximation of a dense set of points needs a parametrization adapted to the degree of the approximating curve while related also to the curve shape. An optimal parametrization can be called more adaptive than intrinsic since it has this combined dependency.

One can observe that designers, by moving interactively poles, have the ability to improve easily over most of the mathematical approaches. Doing so, they produce implicitly the needed adaptive parametrization. An analysis of the designer's actions leads to the definition of an algorithm called "the designer's algorithm". This algorithm produces a parametrization with errors normal to the curve, which is adapted both to the curve shape and the degree of the approximation. A simple mathematical analysis shows that depending on the computation of the poles displacements in the designer's algorithm, the result can minimize the least square distance or a local maxima distance similar to the Tchebychev minimax. With the local maxima distance, the result is nearly equioscillating and improves over the least square distance. The designer's algorithm takes into account easily the extremity conditions needed for CAD data, and extends in a straight forward way to surface approximation.

Our algorithm seems to present practical advantages that can justify a more elaborate mathematical analysis than presented here. Mathematicians may find other and faster ways to produce the adaptive parametrization, and they may also improve the computation of the displacements of poles to accelerate the convergence or produce a real minimax approximation.

References

1. Bézier, P., *Essai de définition numérique de courbes expérimentales*, Thèse de doctorat d'état, Univ. Pierre et Marie Curie, Paris, 02/1977.

2. Brun, J. M., S. Foufou, A. Bouras, and P. Arthaud, In search of an optimal parametrization of curves, in *Proceedings of the fourth Int'l Conf. On computational graphics and visualisation techniques*, Lisbon, Portugal, Dec 11-15, 1995.
3. Bogacki, P., E. W. Stanley, and X. Yuesheng, Degree reduction of Bézier curves by uniform approximation with endpoint interpolation, *Computer-Aided Design* **27,9** (1995), 651-661.
4. Brunnett, G., T. Schreiber, and J. Braun, The geometry of optimal degree reduction of Bézier curves, *Comput. Aided Geom. Design* **13** (1996), 773-788.
5. Eck, M., Degree reduction of Bézier curves, *Comput. Aided Geom. Design* **10** (1993), 237-251.
6. Eck, M., Least square degree reduction of Bézier curves, *Computer-Aided Design* **27,11** (1995), 845-851.
7. Farin, G., *Curves and Surfaces for CAGD a Practical Guide*, Academic Press, 1997.
8. Foufou, S., Brun, J.M., and A. Bouras, Study of iterative processes convergence, in *Proceedings of the third int'l conf. in central europe on computer graphics and visualization*, Plzen, Czech Rep., Feb 14-18, 1995.
9. Harada, K., and E. Nakamae, Application of the Bézier curve to data interpolation, *Computer-Aided Design* **14,1** (1982), 55-59.
10. Hoscheck, J, Intrinsic parametrization for approximation, *Comput. Aided Geom. Design* **5** (1988), 27-31.
11. Lachance, M., Chebyshev economization for parametric surfaces, *Comput. Aided Geom. Design* **5** (1988), 195-208.
12. Lee, E., Choosing nodes in parametric curve interpolation, *Computer-Aided Design* **21,6** (1984), 363-370.
13. Wahu, G., Critical criteria for data reduction in curve and surface models conversion, in *Proceedings of the first Swiss Conference on CAD/CAM*, Neuchâtel, Switzerland, Feb 22-24, 1999.
14. Watkins, M. A., and A. J. Worsey, Degree reduction of Bézier curves, *Computer-Aided Design* **20,7** (1988), 398-405.

G. Wahu and A. Bouras
LIGIM, University Claude Bernard Lyon 1,
43 boulevard du 11 novembre 1918
69622 Villeurbanne, France
gwahu@ligim.univ-lyon1.fr

J.M. Brun
LIM/XAOLab, University Aix-Marseille 2
ESIL, Case 925
13288 Marseille cedex 9, France
jmbrun@esil.univ-mrs.fr