Title:
T-Spline Skinned Surfaces with Feature Curves

Authors:
Ahmad Nasri, anasri.82@gmail.com, Lebanese National Council for Scientific Research
Abdulwahed Abbas, abbas@balamand.edu.lb, The University of Balamand

Keywords:
Skinning, T-spline, Polygonal Complexes, NURBS, Interpolation, Normal and Cross-Curvature.

DOI: 10.14733/cadconfP.2018.397-401

Introduction:
Skinning is a common problem in Computer Graphics and Free-form Surface Design. Typically, it consists of fitting a surface through a pre-defined sequence of curves, known as cross-sections. In the literature, the type of curves ranges from general parametric curves to B-spline curves defined by uniform or non-uniform knot vectors.

This problem was recently addressed in the context of T-spline surfaces generalizing the fitting of such surfaces to cross-sections defined by Non-Uniform B-spline curves with incompatible knot vectors [3]. The resulting surfaces are typically defined by fewer control points than other skinned surfaces.

In this abstract, we report on an extension of T-spline Skinned Surfaces that interpolates curves with cross-derivative information such as normal and curvature values, thus adding more features to these surfaces.

Main Idea:
In a T-spline skinned surface, each curve is defined by a non-uniform B-spline curve with its own knot vector. The union of the knot vectors defining all the cross-sections forms the knot vector of the resulting surface in one direction. Each of the curves, by itself, corresponds to an iso-parametric curve of a certain knot value. The sequence of these knot values corresponds to the knot vector of the surface in the other direction. Typically, the interpolation of the curves generates a system of equations which could be solved locally for each skinned curve.

T-Spline Surfaces
T-spline surfaces are generalization of Non-Uniform Rational B-Spline (NURBS) surfaces [5]. They were introduced by Sederberg et. al. in two pioneering papers [5, 6]. T-spline surfaces inherit most of the properties of NURBS surfaces with an essential additional feature: local refinement.

In T-splines, a new control point could be inserted in the defining mesh without inducing the insertion of a complete row or column as is the case for NURBS surfaces. As such, they are able to represent the same surfaces as NURBS but with fewer control points.

Typically, a T-spline surface is defined by a network of control points called a T-mesh. In a cubic T-spline, which is the one that is the main focus of this paper, a T-mesh topologically mimics its corresponding parameter domain, often referred to as a pre-image; see Fig. 1.

A T-mesh is similar to the grid of control points defining a NURBS surface, except that in a T-mesh, partial rows, or columns, or even isolated control points are allowed. While all control points in the NURBS setting, except the boundary ones, are 4-valent vertices, T-mesh control points could also be 0-valent, 1-valent (referred to as I-junction), 2-valent which could be of two types: V-junction (if connected to two horizontal or two vertical edges), or L-junction (if connected to one horizontal and one vertical edge), 3-valent (referred to as T-junction). In the case where all control points are 4-valent, a T-spline surface degenerates to a NURBS surface.

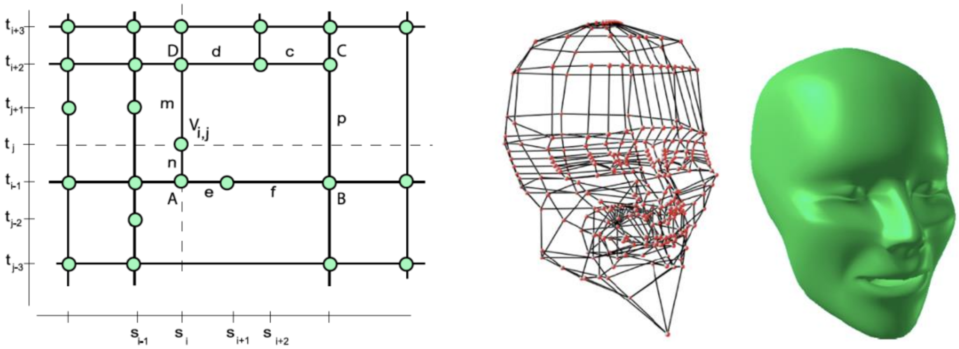

Fig. 1: A T-Spline: (a) T-mesh, (b) Polyhedron, and (c) Surface [3].

A T-spline surface is defined by a T-mesh of control points p_{ij} with some observed rules (see below) and two knot vectors: a horizontal one U and a vertical one V. The expression of a T-spline surface is given by:

$$P(u,v) = \frac{\sum p_{ij} w_{ij} B_{ij}(u,v)}{\sum w_{ij} B_{ij}(u,v)}$$

where each control point p_{ij} is associated with a weight w_{ij} and a blending function $B_{ij}(u,v)$ given by $B_{ij}(u,v) = N_i(u) \times N_j(v)$ where $N_i(u)$ and $N_j(v)$ are the cubic B-spline basis functions defined over a horizontal quintuple knot U_i, and a vertical quintuple knot V_j, respectively.

In the NURBS setting, the knots of the horizontal (vertical) quintuple are typically the two knots before and the two knots after the one at the central position at $< i,j >$.

In the T-spline setting, however, to determine the two quintuples associated with a control point p_{ij}, we consider a horizontal (vertical) ray emanating from that point and consider its intersection with the first two left and two right vertical (horizontal) edges. These intersecting edges determine the knots for the corresponding blending functions.

For example, in Fig. 1, showing a T-spline surface and its corresponding T-mesh, the quintuple corresponding to control point p_{ij} at the knot position $< s_i . t_j >$ are: $U_i = [s_{i-2}, s_{i-1}, s_i, s_{i+2}, s_{i+3}]$, and $V_j = [t_{j-3}, t_{j-1}, t_j, t_{j+2}, t_{j+3}]$. It is common, in a T-mesh, to associate a knot interval with each horizontal (vertical) edge in the mesh. For instance, a horizontal edge connecting two control points p_{ij} and p_{rj} is assigned a knot interval $s_r - s_i$. A knot interval for a vertical edge is similarly devised. In this respect, the following rules are required to ensure the validity of a T-mesh:

1. Two adjacent horizontal or vertical control points on adjacent horizontal or vertical knot lines must be connected by a horizontal or a vertical edge, respectively.
2. The sum of the knot intervals on opposing edges of a face must be equal. For instance, referring to the same Fig. 1 again, the edges of the quad face $ABCD$ should have $e + f = d + c$ and $m + n = p$.

Proceedings of CAD'18, Paris, France, July 9-11, 2018, 397-401

T-Spline Skinned Surfaces

Given a sequence of cubic NURBS curves $(C_i)_{i=0..n}$, as depicted in Fig. 2 below, where each curve is defined by a set of control points (p_{ij}) over a corresponding knot sequence $V_i = (t_j)$, the corresponding T-spline skinned surface is a T-spline surface interpolating these curves and constructed as follows:

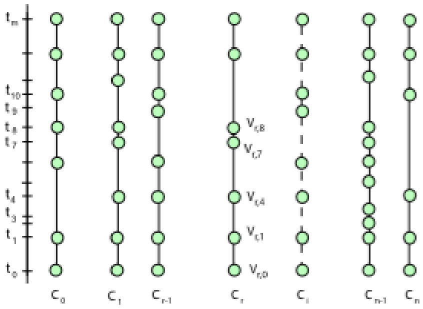

Fig. 2: A set of cubic NURBS curves to be interpolated by a T-spline surface [3].

1. Assign the value 1 as the initial weight of each control point of the preliminary T-mesh P_0.
2. Assign a knot value s_i to each curve C_i to become an iso-parametric curve corresponding to a knot line $s = (s_i)$. The sequence of all knots (s_i) forms the horizontal knot vector U of the skinned surface.
3. Merge all the curve knot vectors (V_i) of the given curves (C_i) into one single knot vector V, which will be used as the vertical knot vector of the skinned surface.
4. Include the edges of the control polygons of the curves C_i in the T-mesh P_0.
5. Include a horizontal edge linking two control points, with the same knot t_j, of every two adjacent curves C_i and C_{i+1}.
6. Insert an additional control point X or Y on every horizontal edge and add a vertical edge between every two adjacent of these additional control points X or Y with the same knot s.
7. For every curve, set up a system of equations to reposition its control points to achieve curve interpolation. The system is solved locally using the intermediate X or Y points as described in [3].

In the above process, skinning is possible for certain configurations with *curve* or *T-junction* vertices.

T-skinning with Features Curves

The skinning routine shown above can be extended to include features on the interpolated curves such as normal, and cross-curvature. Similar to the work reported in [2], we could generate a T-spline skinning surface with cavity or creases, and other features along the interpolated curves. In our discussion, we only consider similar configurations with curve or T-junction vertices as depicted in [3], which are typically the basis for more complicated configurations.

Proceedings of CAD'18, Paris, France, July 9-11, 2018, 397-401

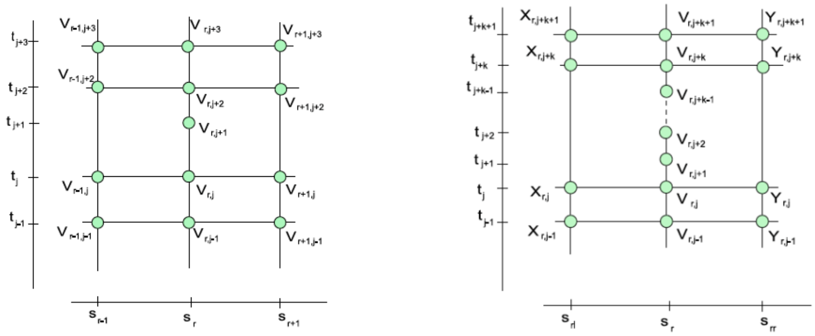

Fig. 3: Normal Interpolation: Configuration of curve vertices [3].

Normal Interpolation
To incorporate surface normal information along a skinned curve, we could formulate our problem as follows (see Fig. 3).

1. Assume that we are given a surface normal n_{ij} at a point V_{ij} of a skinned curve C_i corresponding to a knot line s_i.
2. Let t_{ij} be the tangent direction along the curve at V_{ij}. This is given by $V_{i(j-1)}V_{i(j+1)}$.
3. Let q_{ij} be the cross product of t_{ij} and n_{ij}.
4. Let P_{ij} be the plane formed by the vectors q_{ij} and n_{ij}.
5. We reposition the additional points left $X_{(i-1)j}$ and right $Y_{(i+1)j}$ in the plane P_{ij} along the direction of q_{ij}.

For the point $V_{i(j+1)}$ and similar, we express its corresponding basis functions in terms of those corresponding to the two points $V_{i(j-1)}V_{ij}$ above and the other two points $V_{i(j+2)}V_{i(j+3)}$ below this point.

Curvature Interpolation
Our approach to adding such a feature on the skinned curve is an extension of the algorithm used in [2]. Given a scalar value K associated with a skinned curve, this value is used to reposition the additional X and Y points to control the cross-curvature surface along this skinned curve. In the uniform case, where all knot intervals are equals, the value of the curvature at a knot value of a B-spline curve is given by: $K = 8 \times h / a^2$ [1], where h and a are shown in Fig. 4 below.

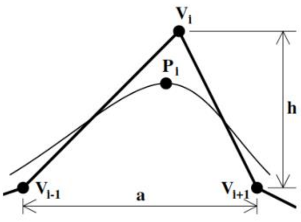

Fig. 4: Cross-Curvature.

The value above is computed from the curvature of the corresponding Bézier curve segment at its end-points. In the T-spline setting, where the surface is non-uniform, the above value needs to be modified to incorporate the possibly unequal knot intervals around the junction points where two curve segments meet. So, our approach consists of the following:

1. Incorporate the knot intervals in the curvature value above.
2. Modify the algorithm in [2] to reposition the triplet ($X_{(i-1)j}$, V_{ij}, $Y_{(i+1)j}$) as follows:
   a. Let T_i be the tangent to the surface along the skinned curve at the limit point corresponding to V_{ij}.
   b. Let Q_i be the plane perpendicular to the curve at the limit point corresponding to V_{ij}.
   c. Let ($Xp_{(i-1)j}$, Vp_{ij}, $Yp_{(i+1)j}$) be the projection of the triple ($X_{(i-1)j}$, V_{ij}, $Y_{(i+1)j}$) into Q_i along the tangent T_i.
   d. Shear ($Xp_{(i-1)j}$, Vp_{ij}, $Yp_{(i+1)j}$) into ($Xt_{(i-1)j}$, Vt_{ij}, $Yt_{(i+1)j}$) to adjust the curvature to K.
   e. Un-project the triple ($Xt_{(i-1)j}$, Vt_{ij}, $Yt_{(i+1)j}$) back into the original plane and along T_i thus giving the final position of the triple ($X_{(i-1)j}$, V_{ij}, $Y_{(i+1)j}$).
   f. In the configuration of curve points, their corresponding basis functions could be incorporated in the formulation of the algorithm.
   g. In the configuration of T-junction points, where either an X or Y point is missing, we express the basis functions of the temporary point, say X', to reposition the other.

Conclusions:
In this abstract, we report on work in progress for adding features along the curves of a T-spline skinned surface. These features are mainly driven by the cross-derivative information along the skinned curves. Normal or tangent planes could be specified along these curves. Furthermore, a scalar value is basically used as simple interface for manipulating the cross-curvature thus modifying the shape of the surface and adding various features along them. These scalar values could be negative or positive generating cavities or bumps, respectively.

Acknowledgement:
The research work reported herein is supported by a grant from the Lebanese Council for Scientific Research, for the academic year 2016-2017.

References:
[1] Farin, G.: Curves and Surfaces for CAGD: a Practical Guide (Fifth Edition), The Morgan Kaufmann Series in Computer Graphics, Elsevier Inc., 2002.
[2] Nasri, A.; Sabin, M.; Zaki, R.-A.; Nassiri, N.; Santina, R.: Feature curves with cross curvature control on Catmull-Clark subdivision surfaces, Advances in Computer Graphics, 2006, 761-768.
[3] Nasri, A.- H.; Sinno, K.; Zheng, J.: Local T-spline surface skinning, The Visual Computer, 28(6-8), 2012, pp. 787-797. http://dx.doi.org/10.1007/s00371-012-0692-1
[4] Piegl, L.; Tiller, W.: The NURBS Book, Second Edition, Springer Verlag, 1997. http://dx.doi.org/10.1007/978-3-642-59223-2
[5] Sederberg, T.-W.; Cardon, D.-L.; Finnigan, G.-T.; North, N.-S.; Zheng, J.; Lyche, T.: T-Spline Simplification and Local Refinement, ACM Transactions on Graphics, 23(3), 2004, pp. 276-283. http://dx.doi.org/10.1145/1015706.1015715
[6] Sederberg, T.-W.; Zheng, J.; Bakenov, A.; Nasri, A.: T-Splines and T-NURCCS, ACM Transactions on Graphics, 22(3), 2003, pp. 477-484. http://dx.doi.org/10.1145/882262.882295