Computer-Aided Design 79 (2016) 36–47

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Reconsideration of T-spline data models and their exchanges using STEP

Wenlei Xiao [a,b], Yazui Liu [a], Rui Li [a], Wei Wang [a], Jianmin Zheng [c], Gang Zhao [a,b,*]

[a] School of Mechanical Engineering & Automation, 100191 Beihang University, Beijing, China

[b] MIIT Key Laboratory of Aeronautics Intelligent Manufacturing, 100191 Beihang University, Beijing, China

[c] School of Computer Engineering, 639798 Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 27 January 2016
Accepted 4 June 2016

Keywords:
T-spline
Data model
STEP
Data exchange

ABSTRACT

T-spline is a new approach to define freeform surfaces with relatively less control points than NURBS and is able to represent a model using a single surface without joining errors. Whereas, the complexity of T-spline data models leads numerous difficulties in its programming, which hinders the research and development of T-spline technologies. In addition, the data exchange of T-spline models still remains on a primitive level, and no standardized data format has been published so far. This article gives a reconsideration to the existing T-spline definitions, and proposes a set of redesigned data models which have much more understanding conveniences to both human and computer. Moreover, STEP-compliant data models are designed using the proposed T-spline models to standardize their data exchange between different CAx systems. The combination of T-spline with other product models in ISO 10303 makes it convenient to exchange the versatile resource data in a hybrid neutral file. A prototype system is developed for the validation purpose, which consists of a TSM-to-STEP convertor, a STEP parser and a T-spline kernel. Using the developed prototype system, one can automatically convert a Rhino system exported TSM file to a STEP file in the P21 format, which can be then parsed using the STEP reader and processed by the T-spline kernel. Some testing examples show that the proposed data models are much more efficient in processing and exchanging the T-spline data.

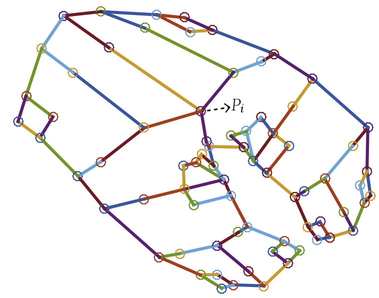
1. Introduction

T-spline has attracted great interests from researchers since its emergency in 2003 [1]. Comparing to NURBS, it has great advantages of less control points, localized refinement and tessellation operations [2,3] and isogeometric analysis [4,5]. In addition, T-spline has shown its progressively powerful modeling functions comparing to NURBS, especially after *Rhino*© releases the T-spline plug-in [6]. The success of T-spline kernel in Rhino has shown an optimistic prospect on integrating T-spline into other CAx systems. For example, a CAD system can provide to the users another modeling method via T-spline, a CAE system can introduce a new basis for isogeometric analysis [7,8], a CAM system may support a novel path planning ability to generate a five-axis machining path for a whole part directly [9], and a CNC system could use a T-spline model as its precisely defined workpiece part for object-oriented and inspection based closed-loop manufacturing [10], which as well obeys the concept of STEP-CNC [11–13]. Therefore, it can be considered that more and more CAx systems will provide this new modeling method in the future. Regarding the prosperous development of T-spline, it will grow up to be a necessity to exchange T-spline models between different CAx systems, just alike the requisite of other conventional B-Rep models. In order to fulfill this request, *Rhino*© has recently unfolded a text-based TSM (T-spline Mesh) file format [14] for storing its exported T-spline data. However, the practical use has proved that the analyticity of TSM is a long way from satisfactory for complex data exchange. Developers generally just have to spend a lot of time and efforts in developing a data parser, before they really can import a T-spline model generated by *Rhino*©. In order to solve this dilemma, the standardization of T-spline models has to be implemented before miscellaneous customized definitions flood the research and development fields.

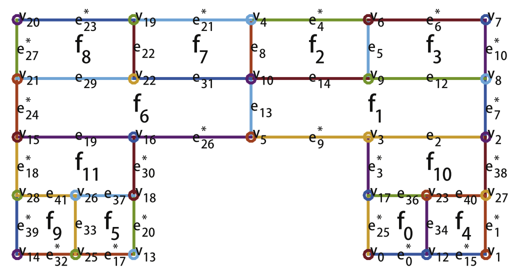
Standardized T-spline models must be compact for storing, flexible for data defining, and reversible for indexing, as data

* Corresponding author at: School of Mechanical Engineering & Automation, 100191 Beihang University, Beijing, China.
E-mail address: zhaog@buaa.edu.cn (G. Zhao).

http://dx.doi.org/10.1016/j.cad.2016.06.004



(a) A T-mesh example.



(b) Pre-image of the T-mesh.

Fig. 1. A T-mesh example and its pre-image.

structures of T-spline are much more complex than that of NURBS. C. Asche et al. proposed a set of efficient data structures for T-spline modeling [15], which rely on the half-edge data structures in the CGAL geometry library [16], whereas the obtained results are still on a primary level, for they still use the conventional two-layer T-spline data models (the T-mesh and its pre-image). Lin et al. introduced the extended T-mesh data structures that make it more easier for computation, but brings many redundant data [17]. Very few more research literatures can be found on the similar topic. The majority of T-spline researchers are obliged to develop their own T-spline kernel from scratch without enough consideration on the efficiency and operability. This status has hindered the development of T-spline for a long time, and drove many new researchers away.

Under those considerations, this paper conducts great efforts on T-spline data modeling using an object-oriented data modeling framework. The conventional T-mesh data structure is decomposed into the parametric, topological and Cartesian layers (so called three-layer models), which have great advantages in storing, accessing and operating the T-spline data. Object-oriented T-spline data models are redesigned to obtain more conveniences to both human and computer, and the STEP-compliant data exchange format is subsequently derived using the EXPRESS modeling language [18], and programmed using the SDAI method [19]. The conventional ISO 10303 standard is thereby extended. Using the STEP standard, NURBS has been successfully modeled, and widely used in the forms of AP203 or AP214 files for CAD data exchanges between different CAx systems. Although STEP has regarded NURBS as the major modeling method for freeform surfaces [20,21], the strive on modeling T-spline using STEP extends the compatibility of the STEP standard to manifold CAD models, let alone the numerous advantages of T-spline against NURBS [1]. If the T-spline models can be supported together with B-Rep models, the data exchange ability can be significantly enhanced and the exchange errors can be managed in a reasonable manner. Since the geometric data models are the base for other applications and implementations, the addition of T-spline extends and changes the way of many other STEP-compliant CAx applications as well.

2. Reconsideration of T-spline data models

This section studies the data structures of T-spline. Firstly, a brief review of T-spline and T-mesh is taken to discuss the complexities and difficulties in handling the existing T-spline data models. Secondly, a set of redesigned T-spline data models are proposed, which organize the T-spline data structures into three layers. The data structures are described and discussed using a simple T-spline example.

2.1. Review of T-spline and T-mesh

Recently, most researches have defined a T-spline surface by means of a control grid called T-mesh [2,22], which provides information in both Cartesian and parametric spaces (control points in the Cartesian space and the pre-image in the (s, t) parametric space). This definition is basically extended from that of the tensor-product B-spline surfaces, which uses a rectangular grid of control points. In order to describe the parametric space, the pre-image of a T-mesh is presented as an attachment to the T-mesh. Fig. 1 shows a so designed T-mesh and its pre-image. These definitions reluctantly work for the theoretical description, whereas bring a lot of troubles in theory understanding, data exchanging and software programming. The difficulties imply that the introduced T-mesh from the control grid of B-spline is not sufficient for many extraordinary properties of T-spline. There are essentially three major drawbacks in the existing T-mesh definitions:

(1) T-mesh contains not only Cartesian but also parametric data, which bring great complexities in operating the data structures.

In order to calculate the equation of a point based T-spline (instead of grid based) [1]

$$P(s,t) = \frac{\sum_{i=1}^{n} P_i B_i(s,t) w_i}{\sum_{i=1}^{n} B_i(s,t) w_i} \tag{1}$$

there are two vital variables that need to be determined in advance: P_i and $B_i(s, t)$. Wherein, P_i are the control points, and $B_i(s, t)$ are the basis functions given by

$$B_i(s,t) = N_{i0}^{m}(s) N_{i0}^{n}(t) \tag{2}$$

where N_{i0}^{m} and N_{i0}^{n} are the mth and nth B-spline basis functions associated with individual knot vectors in s and t directions. Usually $m = n = 3$, so that they become cubic B-splines. Thus, to specify a T-spline, one must provide a pair of knot vectors for each control point.

An intuitive approach is to store a pair of knot vectors in each control point of the T-mesh. However, this approach consumes additional storing space for redundant data, as most knot vector components are the same in adjacent control points. Moreover, the locality of T-spline causes difficulties in determining which control points should be involved in calculating a specified parametric coordinate (s, t). This is because the T-spline equation (Eq. (1)) is point based rather than grid based, so that the involved control points cannot be easily determined from the T-mesh grid, especially when multiple knots occur in the T-mesh.

(2) The complexity of the pre-image of a T-mesh is ignored significantly, although it contains the most abundant information.