Computer-Aided Design 100 (2018) 30–38

Contents lists available at ScienceDirect

Computer-Aided Design

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

Injective hierarchical free-form deformations using THB-splines[☆]

João Pedro Duro Reis [a,*], Jiří Kosinka [b]

[a] Universidade Federal do Rio Grande do Sul, Brazil
[b] University of Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 23 August 2017
Accepted 16 February 2018

Keywords:
Free-form deformation
Model deformation
Truncated hierarchical B-spline
Computer aided design
Injective deformation

ABSTRACT

The free-form deformation (FFD) method deforms geometry in n-dimensional space by employing an n-variate function to deform (parts of) the ambient space. The original method pioneered by Sederberg and Parry in 1986 uses trivariate tensor-product Bernstein polynomials in $\mathbb{R}^3$ and is controlled as a Bézier volume. We propose an extension based on truncated hierarchical B-splines (THB-splines). This offers hierarchical and local refinability, an efficient implementation due to reduced supports of THB-splines, and intuitive control point hiding during FFD interaction. Additionally, we address the issue of fold-overs by efficiently checking the injectivity of the hierarchical deformation in real-time.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

One of the desired features of a modelling software is the capability of deforming an object in an efficient, precise, and smooth manner [1]. This can be achieved through the use of free-form deformation (FFD) techniques. The original FFD method was developed in [2]. It is based on Bernstein polynomials and intuitive control is provided through Bézier volumes. FFDs use the intuition that geometry can be deformed along with the space it is embedded in. This technique is highly flexible as it can be used globally or locally, with any degree of continuity, and even preserve volume [2].

Bernstein polynomials provide a versatile and simple basis for FFDs. However, they suffer from several limitations, most notably they have global support and fixed polynomial degree for a given number of freedoms in the corresponding control structure. This can be alleviated by employing B-splines instead [3]. Moreover, finer control is offered in the rational setting with weights attached to control points [4,5].

FFDs have been further generalised to accommodate deformed control structures [6] and control structures of arbitrary topology [7,8]. Additionally, special techniques have been developed to correctly deform polygonal meshes [9]. In our work, we remain in the structured setting and focus on hierarchical techniques.

Hierarchical splines were introduced in [10] to facilitate local refinement. However, the proposed method does not possess the partition of unity property in the hierarchical setting and thus finer level edits need to be maintained either independently of other levels, or via control vectors [10,11] rather than control points. Hierarchical B-splines have been applied in the context of fitting and image registration [12,13]. Since then, several methods which maintain partition of unity and support local refinement have been proposed. T-splines [14] allow for T-junctions in the control structure, LR-splines [15] rely on local splitting of B-splines, and THB splines [16] restore partition of unity of the hierarchical basis by a truncation mechanism.

Although T-splines have been extended, modified, and even applied in the context of FFDs, see [17,18] and the references therein, their rational nature makes them less efficient than purely piecewise polynomial methods. While LR-splines may seem a good candidate for FFDs as they can be equipped with a control-point structure for the user to manipulate, such control structures typically lack a clear and intuitive hierarchy.

In contrast, THB-splines offer a clear hierarchical structure, which enables us to selectively hide control points from certain levels in the user interface [16,19], and offer polynomial basis functions with reduced supports when compared to non-truncated alternatives [20]. Their locality and numerical efficiency are ideally suited for FEM-based non-rigid image registration [21,22] and for performance-critical scenarios such as real-time FFDs performed on dense meshes. This also allows the use of more demanding techniques such as self-intersection detection and prevention [23] without sacrificing interactivity. Further advantages of B-spline hierarchies are nicely summarised in [19,20]. An example FFD with THB-splines is shown in Fig. 1.

Our main contributions are:

- a real-time FFD method based on THB-splines,
- intuitive user interface with features such as control point hiding and region of influence highlighting,

☆ This paper has been recommended for acceptance by Y.J. Zhang.

* Corresponding author.
E-mail addresses: jpduroreis94@gmail.com (J.P.D. Reis), J.Kosinka@rug.nl (J. Kosinka).

https://doi.org/10.1016/j.cad.2018.02.005
0010-4485/© 2018 Elsevier Ltd. All rights reserved.

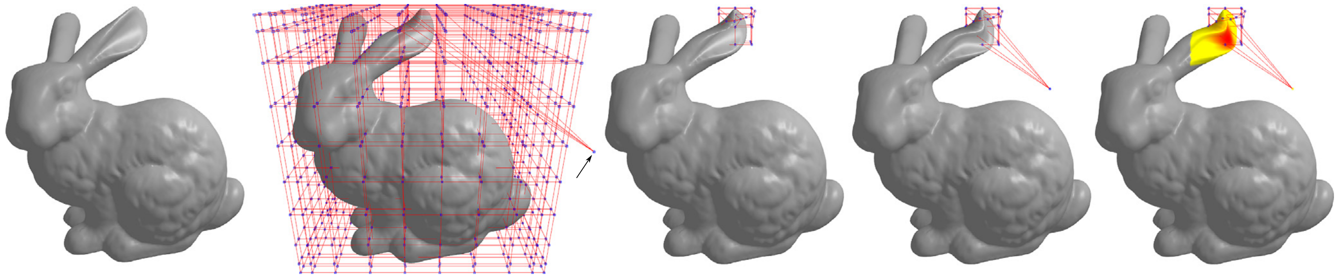

Fig. 1. Far left: The input Stanford Bunny model (35K vertices, 70K faces). Left: The model embedded in a tri-cubic B-spline volume with 10×10×10 control points (level zero), one of which (pointed to by the black arrow) has been moved to adjust the shape of its left ear. Middle: Control structure of level one as requested by the user near the left ear consisting of 3×3×3 control points of level one. All control points of level zero have been on the user's request hidden to avoid visual clutter. Right: One of the control points of level one has been moved to finely adjust the local shape of the ear. Far right: The same situation, but this time with the region of influence of the control point being moved highlighted; yellow indicates small influence and red large influence, as determined by the associated basis function. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

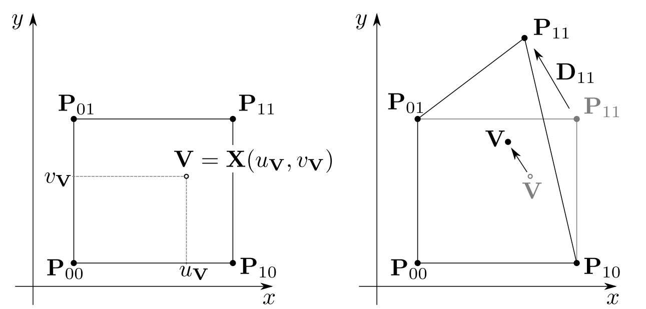

Fig. 2. The FFD concept illustrated on a simple 2D example. Four control points $\mathbf{P}_{ij}$ define the control structure of a bi-linear FFD (left). A single vertex $\mathbf{V}$ with parametric coordinates $(u_\mathbf{V}, u_\mathbf{V})$ is deformed to its new position when $\mathbf{P}_{11}$ is displaced by $\mathbf{D}_{11}$ (right).

- efficient injectivity checking resulting in interactive fold-over prevention.

We start by reviewing FFDs and THB-splines and show how to integrate these two frameworks (Section 2). Then we present our approach to ensuring injectivity of FFDs as well as normal updating (Section 3). Implementation details are presented in Section 4 and results are discussed in Section 5. Finally, we conclude the paper and point to future work (Section 6).

2. Free-from deformations and THB-splines

In this section we recall the basic concepts regarding FFDs and THB-splines.

2.1. Free-form deformations

FFD is a method for deforming objects by moving the control points of a control structure encapsulating (parts of) these objects. There are many approaches to specifying these structures, from regular axis-aligned grids [2] to unstructured and arbitrarily oriented meshes [7]. Once the control structure has been specified, parametric coordinates of each vertex of the deformed object(s), which we assume is a (dense) triangular mesh, are computed. This is in general a difficult problem.

In our work, we assume that the control structure forms a regular tensor-product axis-aligned grid (which can later be refined in a hierarchical manner; see Section 2.2) of control points $\mathbf{P}_{ijk}$ corresponding to a tensor-product B-spline volume of a certain tri-degree. The individual control points are geometrically positioned based on their associated Greville abscissae $\boldsymbol{\xi}_{ijk}$ [24]. In that case, the parametric coordinates $(u_\mathbf{V}, v_\mathbf{V}, w_\mathbf{V})$ of each to-be-deformed vertex $\mathbf{V}$ are easily calculated using a linear transformation [2].

Assume that the associated B-spline volume is given by

$$\mathbf{X}(u, v, w) = \sum_{ijk} \beta_{ijk}(u, v, w)\mathbf{P}_{ijk}, \qquad (1)$$

where β_{ijk} are the tri-variate tensor-product B-splines defined over (typically open-uniform) knot vectors whose size is specified by the number of desired control points in each parametric direction and the spline degree. Then indeed $\mathbf{X}(u, v, w) = (u, v, w)$ when $\mathbf{P}_{ijk} = \boldsymbol{\xi}_{ijk}$ for all i, j, k and thus the parametric coordinates $(u_\mathbf{V}, v_\mathbf{V}, w_\mathbf{V})$ follow from a linear transform. Namely, let $(x_\mathbf{V}, y_\mathbf{V}, z_\mathbf{V})$ be the original coordinates of vertex $\mathbf{V}$, and let $(x_{\min}, y_{\min}, z_{\min})$ and $(x_{\max}, y_{\max}, z_{\max})$ be the minimal and maximal Cartesian coordinates in $\mathbb{R}^3$ among all $\mathbf{P}_{ijk}$, respectively. The parametric coordinates $(u_\mathbf{V}, v_\mathbf{V}, w_\mathbf{V})$ of $\mathbf{V}$ are then

$$u_\mathbf{V} = \frac{x_\mathbf{V} - x_{\min}}{x_{\max} - x_{\min}},$$
$$v_\mathbf{V} = \frac{y_\mathbf{V} - y_{\min}}{y_{\max} - y_{\min}},$$
$$w_\mathbf{V} = \frac{z_\mathbf{V} - z_{\min}}{z_{\max} - z_{\min}}.$$

Once the parametric coordinates of each vertex are known, each $\mathbf{V}$ is mapped (deformed) to $\mathbf{X}(u_\mathbf{V}, v_\mathbf{V}, w_\mathbf{V})$ as the control points $\mathbf{P}_{ijk}$ are moved by the user. An illustration of this concept is shown in Fig. 2.

While this provides a simple and efficient FFD system, local deformations are not possible due to the global tensor-product structure of the involved tri-variate B-splines. In order to support granular deformations in a hierarchical setting, we employ THB-splines.

2.2. Truncated hierarchical B-splines

Truncated hierarchical B-splines [16] provide, as discussed in Section 1, a number of advantages over other hierarchical techniques based on B-splines. We now recall some of their basic properties; further details can be found in [16,20].

Starting from the familiar setting of tensor-product B-splines (1), hierarchical B-splines (HB-splines) first build a hierarchy of these spanning several levels based on a user-specified hierarchy of N nested domains

$$\Omega = \Omega^0 \supseteq \Omega^1 \supseteq \cdots \supseteq \Omega^{N-1} \supseteq \Omega^N = \emptyset. \qquad (2)$$

The auxiliary (empty) set Ω^N is defined to simplify notation below. The level zero domain $\Omega^0 = \Omega$ is the domain of the original tri-variate tensor-product B-splines of (1). An example is shown in Fig. 3.

Download English Version:

https://daneshyari.com/en/article/6876419

Download Persian Version:

https://daneshyari.com/article/6876419

Daneshyari.com