Computer-Aided Design 58 (2015) 173–178

Contents lists available at ScienceDirect

Computer-Aided Design

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

Control vectors for splines☆

Jiří Kosinka [a,*], Malcolm A. Sabin [b], Neil A. Dodgson [a]

[a] Computer Laboratory, University of Cambridge, 15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom
[b] Numerical Geometry Ltd., 19 John Amner Close, Ely, Cambridge CB6 1DT, United Kingdom

HIGHLIGHTS

- We extend traditional splines based on control points by incorporating control vectors.
- Our paradigm allows combining several spline constructions into one formulation.
- We can model curves and surfaces that are not possible with existing techniques.

ARTICLE INFO

Keywords:
Spline
Curve
Surface
Subdivision
Control vector
Modelling

ABSTRACT

Traditionally, modelling using spline curves and surfaces is facilitated by control points. We propose to enhance the modelling process by the use of *control vectors*. This improves upon existing spline representations by providing such facilities as modelling with local (semi-sharp) creases, vanishing and diagonal features, and hierarchical editing. While our prime interest is in surfaces, most of the ideas are more simply described in the curve context. We demonstrate the advantages provided by control vectors on several curve and surface examples and explore avenues for future research on control vectors in the contexts of geometric modelling and finite element analysis based on splines, and B-splines and subdivision in particular.

1. Introduction

Splines have their roots in the lofting technique used in the shipbuilding and aircraft industries throughout the first half of the 20th century. The first mathematical reference to the notion of splines is accredited to the work of Schoenberg [1] on piecewise polynomial approximation. Later, De Casteljau, Bézier, and De Boor [2] contributed invaluably to the development of splines and B-splines in particular; see [3] for the full story.

In the curve case, the modern understanding of a spline can be, in its generality, captured mathematically by

$$\mathbf{c}(t) = \sum_{i=1}^{n} B_i(t)\mathbf{P}_i; \quad t \in [a, b], \tag{1}$$

where $\mathbf{P}_i$ are control points forming a control polygon and $B_i(t)$ form a set of blending functions that satisfy certain properties required by a particular application. To make the spline curve $\mathbf{c}(t)$ well-defined geometrically (i.e., in order to guarantee shape independence of the choice of origin), it is required that $\sum_{i=1}^{n} B_i(t) \equiv 1$ over $[a, b]$. In other words, the blending functions have to form a *partition of unity*. A desired property in some applications (such as finite element analysis) is linear independence of the blending functions; $B_i(t)$ then form a basis and are called basis functions. In the context of analysis, partition of unity can be relaxed to

$$\exists c_i \quad \text{such that} \sum_{i=1}^{n} B_i(t)c_i \equiv 1. \tag{2}$$

A spline curve is generally composed of many pieces of a particular type (e.g., polynomial, rational, trigonometric) joined together at *knots* with a certain continuity. The most popular examples include B-splines [2] (of which Bézier curves are a special case), trigonometric splines [4], interpolating splines [5], and subdivision curves [6].

The above approach can be easily generalised to surfaces. For example, in the tensor-product case we have

$$\mathbf{s}(u, v) = \sum_{i=1}^{n}\sum_{j=1}^{m} B_i(u)B_j(v)\mathbf{P}_{i,j}; \quad (u, v) \in [a, b] \times [c, d], \tag{3}$$

☆ This paper has been recommended for acceptance by Dr. Vadim Shapiro.

* Corresponding author.
E-mail addresses: jiri.kosinka@cl.cam.ac.uk (J. Kosinka), malcolm.sabin@btinternet.com (M.A. Sabin), neil.dodgson@cl.cam.ac.uk (N.A. Dodgson).

http://dx.doi.org/10.1016/j.cad.2014.08.028

where $\mathbf{P}_{i,j}$ form a rectangular control mesh and the univariate blending functions are reused. To be able to cover also triangular patches, volumetric splines, and other flavours of splines including subdivision, we adopt the general notation

$$\mathbf{c}(\mathbf{t}) = \sum_{i \in I} B_i(\mathbf{t})\mathbf{P}_i; \quad \mathbf{t} \in \Omega, \tag{4}$$

where I is an appropriately chosen index set and Ω is a suitable parameter space spanned by $\mathbf{t}$. The functions $B_i(\mathbf{t})$ form a set of blending functions that partition unity or satisfy (2) over Ω.

2. Control vectors for splines

Imagine a scenario where one needs to edit a local detail on a spline, but the blending functions cannot capture it since they are too coarse (with too large a support). Ideally, one would simply like to be able to add a single new control point associated with a blending function of a desired shape and support. However, due to the form of (4) and the partition of unity constraint, adding a new desired blending function is either impossible or difficult as other functions have to be modified as well while keeping the shape of the spline unmodified.

In the case of B-spline curves, one can use knot insertion to refine the space spanned by the B-splines locally. The situation gets much more complicated in the tensor-product surface setting as local refinement is not possible due to the rectangular structure. Several constructions have been proposed to deal with this: T-splines [7], hierarchical B-splines [8,9], truncated B-splines [10], LR B-splines [11], and T-meshes [12]. In these constructions, local refinement is possible as T-junctions are allowed. However, it proved difficult to maintain partition of unity and linear independence for these constructions without imposing restrictions on refinement strategies or moving to the rational setting by normalisation. In the case of THB-splines [10], basis functions are 'truncated' to maintain both properties, but at the expense of introducing functions that may have more than one maximum. While acceptable in analysis, this is undesirable in modelling as a control point's influence is no longer intuitive.

In the context of finite element analysis, (1) was extended to

$$\mathbf{c}(t) = \sum_{i=1}^{n} B_i(t)\mathbf{P}_i + \sum_{i=1}^{n} f(t)B_i(t)\mathbf{V}_i \tag{5}$$

with some function $f(t)$ well suited for a particular application/problem (a typical example is $f(t) = e^t$) and both $\mathbf{P}_i$ and $\mathbf{V}_i$ are treated as degrees of freedom. This approach and its generalisations were introduced in [13,14], and called Partition of Unity Finite Elements and Extended Finite Elements, respectively. In the latter case, the modification was motivated by introducing cracks ($f(t)$ would be a discontinuous function) without having to remesh. Recently, due to the popularity of Isogeometric Analysis (IgA for short; see [15]), such modifications are ever more important.

These modifications, however, break the partition of unity property and cannot be used for modelling directly, as also required by IgA. To amend the situation and to address the requirements in modelling and analysis, we propose to generalise (4) and (5) to

$$\mathbf{c}(\mathbf{t}) = \sum_{i \in I} B_i(\mathbf{t})\mathbf{P}_i + \sum_{j \in J} C_j(\mathbf{t})\mathbf{V}_j; \quad \mathbf{t} \in \Omega, \tag{6}$$

where $\sum_{i \in I} B_i(\mathbf{t}) \equiv 1$ still holds and $\mathbf{P}_i$ are control points, but $\mathbf{V}_j$ are understood as *control vectors*. While this may seem as semantics only, the transition from control points to control vectors allows the functions $C_j(\mathbf{t})$ to be incorporated in the spline definition; the control vectors do not transform as points, but as displacements. Thus, the partition of unity property no longer applies to the set of $C_j(\mathbf{t})$. However, since (2) still applies, this is not a problem for analysis.

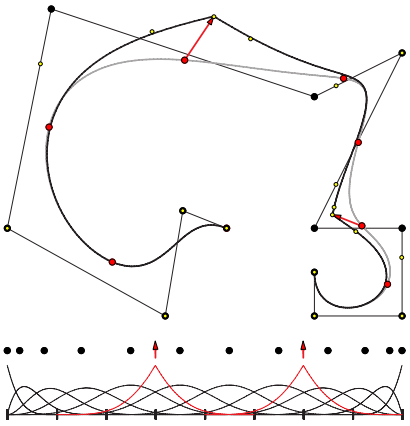

Fig. 1. A degree four spline curve with 12 control points and two non-zero control vectors (red arrows). The associated basis functions are shown in black. The unmodified underlying spline is shown in grey for reference. Control vectors are logically associated with knots. To express the same curve using standard quartic B-splines would require 18 control points (yellow). Each non-zero control vector would give rise to $d - 1 = 3$ extra control points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Considering a spline as a sum of weighted points plus a sum of weighted control vectors, (6), is a paradigm that opens up a wealth of possibilities with minimal increased cost, as we demonstrate below. For future use, we denote $\mathbf{B}$ the collection of B_i, $i \in I$, and $\mathbf{C}$ the collection of C_j, $j \in J$.

In modelling and other applications, one would associate control vectors with desired blending functions and set their magnitudes to zero. This guarantees that the underlying spline given by $\sum_{i \in I} B_i(\mathbf{t})\mathbf{P}_i$ is recovered and the user (or an error estimator in the case of analysis) can then adjust control vectors to fine-tune the resulting shape or approximation. Moreover, several levels of control vectors can be added to obtain a hierarchical structure that facilitates multiresolution editing; see Fig. 4 for an example. For visualisation and demonstrative purposes, we focus on (local) sharp creases and multiresolution editing throughout this paper, but it should be emphasised that the full scope of our paradigm based on control vectors is not limited to these.

In modelling, the convex hull property is advantageous in some situations. In analysis, functions in $\mathbf{B}$ and $\mathbf{C}$ are required to be linearly independent. However, since the form of (6) is very general, the questions of linear independence and convex hulls need to be investigated on a case-by-case basis.

While control vectors can be applied in any spline scenario covered by (6), we focus only on the most important families used in modelling, animation, and IgA: B-splines (Sections 3 and 5) and subdivision based on B-splines (Section 4).

3. Control vectors for B-spline curves

We now investigate (6) for the case of B-spline curves. In the univariate case, we have $\mathbf{t} = t$ and $\Omega = [a, b]$ in (6). We specialise the basis functions to B-splines [2], but any type of splines can be employed (polynomial, rational, trigonometric, etc.) with $\mathbf{B}$ and $\mathbf{C}$ from the same or different families. While the degrees of $\mathbf{B}$ and $\mathbf{C}$ can be in general different, it is sensible to assume that both sets consist of B-splines of one degree d. Similarly, the knots of $\mathbf{B}$ and $\mathbf{C}$ may be completely unrelated, but it is reasonable to consider only cases where at least some of the knots are aligned.

Fig. 1 shows an example of a degree 4 spline curve, where the knots of $\mathbf{B}$ and $\mathbf{C}$ are shared. The B_i are degree four B-splines