Computer Aided Geometric Design 44 (2016) 10–14

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Short Communication

Repeated local operations for m-ary $2N$-point Dubuc–Deslauriers subdivision schemes ☆

Chongyang Deng [a], Yajuan Li [a], Huixia Xu [b,*]

[a] School of Science, Hangzhou Dianzi University, Hangzhou 310018, China
[b] Institute of Mathematics, Zhejiang Wanli University, Ningbo 315100, China

ARTICLE INFO

Article history:
Received 30 October 2015
Received in revised form 31 March 2016
Accepted 11 April 2016
Available online 22 April 2016

Keywords:
m-Ary Dubuc–Deslauriers interpolatory subdivision
Repeated local operations

ABSTRACT

We propose to implement the m-ary $2N$-point Dubuc–Deslauriers subdivision scheme (DDSS) using a series of repeated local operations, which are based on a recursive formula between the newly inserted points of m-ary $2N$-point DDSS and those of m-ary $(2N-2)$-point and m-ary $(2N-4)$-point DDSSs. Numerical analysis reveals the robustness of our method.

1. Introduction

Let $\{\boldsymbol{P}_{0,n} \in \mathbb{R}^d\}$ be the initial control points and $\{\boldsymbol{P}_{K,n}\}(K \geq 1) \in \mathbb{R}^d$ be the refined control points at level K, then the m-ary $2N$-point Dubuc–Deslauriers subdivision scheme (DDSS) is iteratively defined as follows (Deslauriers and Dubuc, 1989)

$$\boldsymbol{P}_{K+1,mn} = \boldsymbol{P}_{K,n}, \tag{1}$$

$$\boldsymbol{P}_{K+1,mn+k} = \mathcal{P}^N_{K,n}\left(\frac{k}{m}\right), k = 1, 2, \cdots, m-1, \tag{2}$$

where

$$\mathcal{P}^N_{K,n}(t) = \sum_{j=-N+1}^{N} L^N_j(t)\boldsymbol{P}_{K,n+j}$$

is the $(2N-1)$-degree polynomial curve interpolating the points $\{\boldsymbol{P}_{K,j}\}_{j=n-N+1}^{n+N}$ at knots $\{-N+1, -N+2, \cdots, N\}$, and

☆ This paper has been recommended for acceptance by Hartmut Prautzsch.

* Corresponding author.
E-mail addresses: dcy@hdu.edu.cn (C. Deng), liyajuan@hdu.edu.cn (Y. Li), xuhx0916@hotmail.com (H. Xu).

http://dx.doi.org/10.1016/j.cagd.2016.04.001
0167-8396/

$$L_j^N(t) = \frac{\prod_{i=-N+1, i\neq j}^{N} (t-i)}{\prod_{i=-N+1, i\neq j}^{N} (j-i)} \tag{3}$$

is the j-th Lagrange basis function defined on knots $\{-N+1, -N+2, \cdots, N\}$.

Generally, the computation of high-degree polynomials is not very robust and efficient, so is that of the new inserted points of m-ary $2N$-point DDSS for large N, which are sampled from a polynomial with degree $2N-1$. Furthermore, it should store many mask coefficients in case of large values of N.

In the field of geometric modeling, pyramid algorithms (Goldman, 2002; Hormann and Schaefer, 2016) are widely used to avoid the inefficiency and instability of computing high-degree polynomials. Analogues of pyramid algorithms in subdivision are the subdivision algorithms using repeated local operations, in which only neighboring information is included. For example, repeated local operations are used to implement approximating subdivision schemes generalized from high-degree splines (Lane and Riesenfeld, 1980; Stam, 2001; Zorin and Schröder, 2001; Cashman et al., 2009), interpolatory subdivision for quadrilateral meshes (Deng and Ma, 2013) and pseudo-spline subdivision surfaces (Deng and Hormann, 2014). In this note, we derive repeated local operations for m-ary $2N$-point DDSS in Section 3 based on the recursive formula given in Section 2.

2. Recursive formula for the coefficients of m-ary $2N$-point DDSS

Theorem 2.1. *Let $L_j^{M-1}(t)$, $L_j^M(t)$ and $L_j^{M+1}(t)$ be the j-th Lagrange basis functions defined on knots $\{-M+2, \cdots, M-1\}$, $\{-M+1, \cdots, M\}$ and $\{-M, \cdots, M+1\}$, respectively, then for $-M \le j \le M+1$, we have*

$$\begin{aligned} & L_j^{M+1}\left(\frac{k}{m}\right) - L_j^M\left(\frac{k}{m}\right) \\ & = \frac{(Mm-k)(Mm+k-m)}{m^2(4M^2-1)}\left[D_j^M\left(\frac{k}{m}\right) + \frac{1}{2M}D_j^M\left(1-\frac{k}{m}\right)\right], \end{aligned} \tag{4}$$

where

$$D_j^M(t) = 2\left[L_j^M(t) - L_j^{M-1}(t)\right] - \left[L_{j-1}^M(t) - L_{j-1}^{M-1}(t)\right] - \left[L_{j+1}^M(t) - L_{j+1}^{M-1}(t)\right], \tag{5}$$

$$L_j^M\left(\frac{k}{m}\right) = 0 (j=-M, M+1), L_j^{M-1}\left(\frac{k}{m}\right) = 0 (j=-M, -M+1, M, M+1),$$

and $m > k > 0$ are two integers.

Proof. For $-M \le j \le -M+3$ and $M-2 \le j \le M+1$, straightforward computations yield Eq. (4). Hence only the cases of $-M+4 \le j \le M-3$ should be verified.

First, substituting $\frac{k}{m}$ for t in $L_j^{M+1}(t)$ and $L_j^M(t)$ leads to the fact that

$$\begin{aligned} & L_j^{M+1}\left(\frac{k}{m}\right) - L_j^M\left(\frac{k}{m}\right) \\ & = \left[\left(\frac{k}{m} - M - 1\right)\left(\frac{k}{m} + M\right) - (j-M-1)(M+j)\right] \cdot \frac{\prod_{i=-M+1, i\neq j}^{M} \left(\frac{k}{m} - i\right)}{\prod_{i=-M, i\neq j}^{M+1} (j-i)} \\ & = \frac{(k-mM-m)(k+mM) + m^2(M+1-j)(M+j)}{m(k-mj)(-1)^{M+1-j}(M+j)!(M+1-j)!} \cdot \prod_{i=-M+1}^{M}\left(\frac{k}{m} - i\right) \\ & = \frac{(-1)^{M+1-j}}{(M+j)!(M+1-j)!}\left(\frac{k}{m} + j - 1\right) \cdot \prod_{i=-M+1}^{M}\left(\frac{k}{m} - i\right). \end{aligned} \tag{6}$$

Next, by the definition (5) of $D_j^M\left(\frac{k}{m}\right)$, it can be seen that