Computers & Graphics 35 (2011) 967–975

Contents lists available at ScienceDirect

Computers & Graphics

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

Technical Section

An extended iterative format for the progressive-iteration approximation ☆

Hongwei Lin *, Zhiyu Zhang

State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058, China

ARTICLE INFO

Article history:
Received 27 December 2010
Received in revised form
16 July 2011
Accepted 18 July 2011
Available online 28 July 2011

Keywords:
Progressive-iteration approximation
Data fitting
Progressive interpolation
Geometric design
Parallel computing

ABSTRACT

Progressive-iteration approximation (PIA) is a new data fitting technique developed recently for blending curves and surfaces. Taking the given data points as the initial control points, PIA constructs a series of fitting curves (surfaces) by adjusting the control points iteratively, while the limit curve (surface) interpolates the data points. More importantly, progressive-iteration approximation has the local property, that is, the limit curve (surface) can interpolate a subset of data points by just adjusting a part of corresponding control points, and remaining others unchanged. However, the current PIA format requires that the number of the control points equals that of the data points, thus making the PIA technique inappropriate to fitting large scale data points. To overcome this drawback, in this paper, we develop an *extended PIA* (EPIA) format, which allows that the number of the control points is less than that of the given data points. Moreover, since the main computations of EPIA are independent, they can be performed in parallel efficiently, with storage requirement $O(n)$, where n is the number of the control points. Therefore, due to its local property and parallel computing capability, the EPIA technique has great potential in large scale data fitting. Specifically, by the EPIA format, we develop an incremental data fitting algorithm in this paper. In addition, some examples are demonstrated in this paper, all implemented by the *parallel computing toolbox* of Matlab, and run on a PC with a four-core CPU.

1. Introduction

Data fitting is a fundamental tool in solving scientific and engineering problems found in the real world, which constructs a curve or patch, or a mathematical function, that has the best fit to a series of data points, possibly subject to constraints. A desirable data fitting method should be able to control the fitting precision for each data point individually. Thus, the data points can be fitted adaptively, that is, only the data points with dissatisfied precision need to be dealt with. In this manner, the computation resource can be saved greatly, especially in fitting large scale data points. However, traditional methods for data fitting usually solve a linear system, hence it is impossible to control the fitting precision for each data point individually.

Moreover, in fitting large scale data points, the incremental manner is often employed, which starts from an initial fitting curve. If the precision of the current fitting curve does not meet the requirement, the number of the control points of the fitting curve is increased, and the data points are fitted again, improving the fitting precision. The desirable manner for re-fitting the data points is able to take advantage of the last fitting result, thus saving the computational resource. However, when we re-fit the data points using the conventional method, such as the least square fitting technique, the last fitting result is totally discarded.

Recently, a new data fitting technique, *progressive-iteration approximation* (PIA) is presented. The PIA method is an iterative procedure, starting by an initial blending curve with control points and blending basis (it also works for tensor product surfaces, we omit the surface case for brevity here), and the limit curve interpolates the given data points. In each iteration, the main computation is to evaluate the foot points corresponding to the given data points. The point evaluation in each iteration is independent completely, so it is suitable to be calculated in parallel in its nature. Moreover, it has been shown that the PIA method has the local property, that is, it can control the fitting precision for each data point separately. In addition, while it requires $O(n^2)$ memory to solve the linear system for data fitting, where n is the number of unknowns, the PIA method decreases the storage requirement to $O(n)$.

However, in the classical PIA format, the number of the control points is equal to that of the data points. It is not feasible when the number of data points is very large. In this paper, we present an *extended PIA* (EPIA) format, where the number of the control points is less than that of the data points. Together with the local property and parallel computing capability of PIA, the extended

☆ This article recommended for publication by Jun-Hai Yong.

* Corresponding author.
E-mail address: hwlin@cad.zju.edu.cn (H. Lin).

0097-8493/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2011.07.003

PIA is a desirable tool for fitting large scale data. By the EPIA format, we develop an incremental data fitting algorithm in this paper, which has the following advantages:

- it can control the fitting precision to each data point individually;
- in fitting data points incrementally, the new fitting procedure starts from the last fitting result.

This paper is organized as follows. In Section 1.1, we review the related work briefly. Section 2 overviews the classical PIA method presented previously. In Section 3, we present the extended PIA method, and show its convergence. Moreover, in Section 4, we develop the incremental data fitting algorithm by the EPIA format. Section 5 demonstrates some examples, all implemented by the *parallel computing toolbox* of Matlab. Finally, Section 6 concludes the paper.

1.1. Related work

The *progressive-iteration approximation* (PIA) is a new technique to seek the curve or patch fitting the data points. The PIA property of the uniform cubic B-spline curve, first discovered by Qi et. al. [1] and de Boor [2], respectively, generates a sequence of curves by adjusting the control points iteratively, and the limit curve interpolates the given data points. In Ref. [3], Lin et al. show that the nonuniform cubic B-spline curve and surface also hold the property. Furthermore, the PIA method is extended to the blending curve and surface with normalized totally positive basis [4]. In Ref. [5], the convergence rates of different bases are compared, and the basis with the fastest convergence rate is found. Moreover, it is proved that the rational B-spline curve and surface (NURBS) have the property, too [6]. Recently, Martin et al. [7] devise an iterative format for fitting, which is actually the progressive-iteration approximation (PIA) format for the uniform periodic cubic B-spline. Very recently, Lu [8] devises a weighted PIA format to speed up the convergence of the PIA method. More importantly, Lin [9] discovers the local property of the PIA, by which PIA can control the fitting precision of each data point individually.

While the PIA format depends on the parametric distance between the data points and the corresponding foot points on the curve with the same parameters, Maekawa et al. invent an iterative fitting format, called *interpolation by geometric algorithm* [10,11], which is similar to PIA format, but relies on geometric distance between the data points and their closest points on the curve. Lin [12] shows the convergence of the interpolation by geometric algorithm. Moreover, the geometric interpolation algorithm [10] is extended to approximate the vertices of a triangular mesh using Loop subdivision surface [13]. On the other hand, the squared distance minimization method [14,15] is also an iterative data fitting algorithm.

Furthermore, the PIA format has been extended to subdivision surface fitting, named *progressive interpolation* (PI). Cheng et al. design the PI format of subdivision fitting for Loop subdivision surface [16,17], and prove its convergence. Fan et al. develop the PI format of Doo–Sabin subdivision surface fitting [18]. The PI format for Catmull–Clark subdivision surface fitting is proposed in Ref. [19].

2. Overview of the classical progressive-iteration approximation

The classical PIA method begins with an initial blending curve or patch with normalized totally positive basis, and generates a sequence of curves or patches by adjusting the control points iteratively.

Specifically, given an ordered data point set

$$\{\boldsymbol{P}_0,\boldsymbol{P}_1,\ldots,\boldsymbol{P}_n\},$$

each point is assigned a parameter $t_i, i=0,1,\ldots,n$, where $t_0<t_1<\cdots<t_n$. Taking these data points as the initial control points, that is, $\boldsymbol{P}_i^0=\boldsymbol{P}_i, i=0,1,\ldots,n$, the initial blending curve $\boldsymbol{P}^0(t)$ can be constructed as

$$\boldsymbol{P}^0(t)=\sum_{i=0}^{n}\boldsymbol{P}_i^0 B_i(t), \tag{1}$$

where $\{B_i(t);\ i=0,1,\ldots,n\}$ is the normalized totally positive blending basis.

Suppose the kth curve $\boldsymbol{P}^k(t)$ has been generated. By computing the foot points $\boldsymbol{P}^k(t_i)$ on the kth curve $\boldsymbol{P}^k(t)$, constructing the *difference vectors* $\boldsymbol{\Delta}_i^k=\boldsymbol{P}_i-\boldsymbol{P}^k(t_i)$, and moving the control points along the corresponding difference vectors, that is, $\boldsymbol{P}_i^{k+1}=\boldsymbol{P}_i^k+\boldsymbol{\Delta}_i^k, i=0,1,\ldots,n$, we get the next curve $\boldsymbol{P}^{k+1}(t)$,

$$\boldsymbol{P}^{k+1}(t)=\sum_{i=0}^{n}\boldsymbol{P}_i^{k+1}B_i(t). \tag{2}$$

In this way, a sequence of curves, $\{\boldsymbol{P}^k(t);\ k=0,1,\ldots\}$, are generated. It has been proved in Refs. [3,4] that the limit curve of the sequence interpolates the data points, i.e., $\lim_{k\to\infty}\boldsymbol{P}^k(t_i)=\boldsymbol{P}_i, i=0,1,\ldots,n$, if the blending basis $\{B_i(t);\ i=0,1,\ldots,n\}$ is normalized totally positive, and its collocation matrix on $t_0,t_1,\ldots,t_n$ is nonsingular.

Similarly, PIA can also be employed to fit a patch to a data array. Readers can refer to Refs. [3,4] for more details.

More importantly, the PIA technique has the local property. That is, if only a subset of the control points are adjusted iteratively, and the others remain unchanged, the limit curve will interpolate the subset of data points, corresponding to the adjusted control points [9]. The local property of the PIA implies that, the fitting precision of each data point can be controlled separately. As a result, the data points can be fitted adaptively [9].

It can be seen from the generation of the curve sequence $\{\boldsymbol{P}^k(t), k=0,1,\ldots\}$ that, in each iteration, the computation of the foot points, the construction of the difference vectors, and the movement of the control points are fully independent, so it is well suited to be implemented by parallel computing.

3. The extended PIA format and its convergence

As stated above, in the classical PIA, the number of the control points of the blending curve or patch is equal to that of the given data points. It is not feasible in fitting a large number of data points. Therefore, in this paper, we develop an extended PIA (EPIA) format, and show its convergence, where the number of the control points is less than that of the data points.

3.1. The extended PIA for curve fitting

Given a data point sequence,

$$\{\boldsymbol{P}_j; j=0,1,\ldots,n\}, \tag{3}$$

after parametrization, each data point $\boldsymbol{P}_j$ is assigned a parameter $t_j, j=0,1,\ldots,n$ with $t_j<t_{j+1}$.

Suppose the initial curve is

$$\boldsymbol{C}^0(t)=\sum_{i=0}^{l}\boldsymbol{C}_i^0 B_i(t), \tag{4}$$

where $\boldsymbol{C}_i^0,\ i=0,1,\ldots,l$, are the initial control points selected from the data points, and $B_i(t), i=0,1,\ldots,l$, are the normalized totally