



Functional data approximation on bounded domains using polygonal finite elements [☆]



Juan Cao^{a,b}, Yanyang Xiao^c, Zhonggui Chen^{c,*}, Wenping Wang^d,
Chandrajit Bajaj^e

^a School of Mathematical Sciences, Xiamen University, Xiamen, 361005, China

^b Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computation, Xiamen University, Xiamen, 361005, China

^c Department of Computer Science, Xiamen University, Xiamen, 361005, China

^d Department of Computer Science, The University of Hong Kong, 999077, China

^e Department of Computer Science, University of Texas at Austin, Austin, TX, 78712, USA

ARTICLE INFO

Article history:

Received 1 May 2017

Received in revised form 2 April 2018

Accepted 15 May 2018

Available online 18 May 2018

Keywords:

Polygonal elements

Barycentric coordinates

Data approximation

Voronoi tessellation

ABSTRACT

We construct and analyze piecewise approximations of functional data on arbitrary 2D bounded domains using generalized barycentric finite elements, and particularly quadratic serendipity elements for planar polygons. We compare approximation qualities (precision/convergence) of these partition-of-unity finite elements through numerical experiments, using Wachspress coordinates, natural neighbor coordinates, Poisson coordinates, mean value coordinates, and quadratic serendipity bases over polygonal meshes on the domain. For a convex n -sided polygon, the quadratic serendipity elements have $2n$ basis functions, associated in a Lagrange-like fashion to each vertex and each edge midpoint, rather than the usual $n(n+1)/2$ basis functions to achieve quadratic convergence. Two greedy algorithms are proposed to generate Voronoi meshes for adaptive functional/scattered data approximations. Experimental results show space/accuracy advantages for these quadratic serendipity finite elements on polygonal domains versus traditional finite elements over simplicial meshes. Polygonal meshes and parameter coefficients of the quadratic serendipity finite elements obtained by our greedy algorithms can be further refined using an L_2 -optimization to improve the piecewise functional approximation. We conduct several experiments to demonstrate the efficacy of our algorithm for modeling features/discontinuities in functional data/image approximation.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

The problem of approximating/fitting given data by smooth or piecewise-smooth bivariate functions has been extensively studied in fields such as computer aided geometric design, computer graphics and computer vision (Powell, 1981). There are a great variety of bivariate functions used in data approximation, such as power functions (Fedele and Ferrise, 2011), various splines (Greiner and Hormann, 1996; Cao et al., 2012; Lai and Schumaker, 2007), and radial basis functions (Carr et

[☆] This paper has been recommended for acceptance by Michael Floater.

* Corresponding author.

E-mail address: chenzhonggui@xmu.edu.cn (Z. Chen).

al., 2001). The appropriate choice of bivariate function spaces for the purpose of data approximation depends on the characteristics of the given input data. For example, input data acquired from smooth surfaces are expected to be approximated by globally smooth functions or piecewise splines with in-built smoothness control, such as B-splines (Mann, 2006) and DMS-splines (Dahmen et al., 1992). Piecewise continuous surfaces that have sharp curvilinear features are expected to be approximated by a set of piecewise smooth functions which can be defined on disjoint regions independently. The focus of our paper is in constructing such functional data approximations, using smooth polygonal finite elements, with adaptation to data features.

A fundamental problem in piecewise smooth approximation is to appropriately decompose the parametric domain into a modest number of sub-regions, such that the local and global approximation reaches a pre-specified error threshold. Besides tensor product splines, which are limited to work on rectangular domains, there are a large number of existing methods utilizing piecewise polynomials on triangulations of the domain which interpolate or approximate input data (Lai and Schumaker, 2007; Lai, 2008). Furthermore, significant effort has been devoted to constructing approximations on triangulations, with continuity of higher order across triangle boundaries (Powell and Sabin, 1977; Mann, 1999) and (or) with higher approximation orders (Zhou and Lai, 2013; Zhou et al., 2008; von Golitschek et al., 2002). In certain applications, the lack or presence of appropriate approximation precision is most important. For example, linear functions on triangulations are commonly used in image data approximation. Since the optimal solution of image data approximation by piecewise smooth functions is generally unavailable, a large number of previous papers focus on finding sub-optimal solutions and in a heuristic manner. Basically, these methods may proceed in either of the following two ways: start with a sparse triangulation and adaptively refine the triangulation to lower the approximation error (Sarkis and Diepold, 2009; Li and Adams, 2013; Adams, 2011, 2013); start from a dense triangulation and gradually decrease the number of vertices to satisfy a pre-specified error threshold (Iske and Demaret, 2015). However, most mentioned methods suffer from too many decomposed patches, and low approximation precision within each patch.

Of course, the decomposed sub-regions or the support of the patches do not need to be triangles or rectangles, and can have a patch function basis yielding high approximation orders. In this context, the approximation problem becomes more complicated and achieving optimal approximations is still actively being pursued (Amirfakhrian, 2010; Fedele and Ferrise, 2011; Dunkl, 1987; Farouki et al., 2003). Voronoi tessellations are perhaps the most popular polygonal tessellations used for piecewise approximation, due to their elegant geometric properties and duality with Delaunay triangulations. There are several papers in the numerical computation and PDE communities which use Voronoi tessellations to discretize the simulation domain, and use generalized barycentric coordinates (GBCs) (Floater, 2015) or their higher order generalizations (Rand et al., 2014). These papers utilize relatively uniform Voronoi tessellations to achieve numerical robustness of finite element computations. For example, the maximal Poisson-sampling method (Sukumar, 2013), the centroidal Voronoi diagram method (Talischi et al., 2010), and the short-edge removing method (Sukumar and Tabarraei, 2004) were used for generating Voronoi meshes that are uniform in size. Instead of using uniform Voronoi diagrams, the focus of recent attempts has been on generating non-uniform Voronoi or optimized diagrams to achieve adaptive piecewise approximations. In particular, optimization methods were proposed in Nivoliers and Lévy (2013) and Chen et al. (2014) to generate non-uniform Voronoi tessellations adapting to the features of the target functions or images, where the features or discontinuities were well preserved by aligning the Voronoi cells along them. However, these methods find it difficult to achieve geometric continuities across cell boundaries as they use full power order polynomials on each Voronoi cell. In this paper, while we also focus on piecewise approximation on arbitrary Voronoi diagrams, in contrast to previous methods, we resort to generalized barycentric coordinates based quadratic serendipity element (QSE) bases (Rand et al., 2014). One is able to achieve continuous approximations over non-uniform Voronoi meshes, while obtaining superior trade-offs between approximation precision and the number of Voronoi cells. The proposed method can additionally be applied to a function that has discontinuities.

On a convex n -sided polygon, QSEs have $2n$ basis functions, associated in a Lagrange-like fashion to each vertex and each edge midpoint (Rand et al., 2014). QSEs have many attractive theoretical properties, such as interpolation, smoothness, and quadratic polynomial reproducibility, therefore they are ideal for the purpose of piecewise continuous approximation. For example, benefiting from the interpolation property of QSEs, the approximation with a quadratic convergence rate on each sub-region and the C^0 continuity between adjacent patches can be directly obtained without solving any linear system with respect to the coefficients associated with the bases. In this paper, we verify the theoretical properties of quadratic serendipity bases by numerical experiments and propose two polygonal mesh generation methods for them to better approximate functional and scattered data (including triangular meshes and images). Our paper's specific contributions include:

1. We construct and analyze various trade-offs in piecewise approximations of functional data, discrete triangular mesh surfaces and images, using generalized barycentric finite elements and particularly quadratic serendipity elements for planar polygons. Moreover, we compare the approximation qualities (precision/convergence) of these partition-of-unity, generalized barycentric finite elements, through several numerical experiments, and using various barycentric coordinates. These are Wachspress coordinates (Wachspress, 1975), mean value coordinates (MVCs) (Hormann and Floater, 2006), natural neighbor coordinates (also called as Sibson's coordinates) (Sibson, 1980), and Poisson coordinates (Li and Hu, 2013).
2. Two greedy algorithms to generate Voronoi meshes for adaptive functional/scattered data approximation. The results show space/accuracy advantages for the quadratic finite elements on these Voronoi polygonal domains versus traditional finite elements over simplicial meshes.

3. The Voronoi polygonal meshes and parameter coefficients of generalized barycentric coordinates based quadratic serendipity finite elements are further refined using an L_2 -optimization method to improve the piecewise approximation. Our results thereby achieve a better trade-off between the approximation error and the number of Voronoi cells that cover the data domain. Additionally, our experimental results show the efficacy of our algorithms for modeling features/discontinuities in functional data/image approximation.

The rest of the paper is organized as follows. In Section 2, we review the definition of quadratic serendipity elements for convex planar polygons. In Section 3, we compare approximation qualities (precision/convergence) of barycentric coordinates on Delaunay triangulations/Voronoi diagrams and quadratic serendipity elements on Voronoi diagrams. Two greedy algorithms are proposed to generate Voronoi diagrams which adapt to the distributions of the given data. In Section 4, we further improve the Voronoi diagrams by optimizing a tailored objective function such that approximation results better recover the features in the given data. In Section 5, a mesh optimization method is applied to image approximation, and we show various examples by fitting quadratic serendipity elements to different images. Finally, we present our conclusions and directions for further work in Section 6.

2. Quadratic serendipity elements for convex planar polygons

Generalized barycentric coordinates on general polygons are a convenient tool to linearly interpolate data that is given on a polygonal domain. Their widespread usage ranges from computer graphics through to computational mathematics. In particular, generalized barycentric coordinates span the space of linear polynomials on general polygons. Hence, they are considered as the shape functions for polygonal finite elements and utilized in various applications (Floater and Lai, 2016; Gillette and Bajaj, 2011; Martin et al., 2008; Milbradt and Pick, 2008; Rashid and Selimotic, 2006; Gillette et al., 2012; Sukumar and Malsch, 2006; Sukumar and Tabarraei, 2004; Tabarraei and Sukumar, 2006; Wicke et al., 2007). Despite their varieties, only a first order of approximation can be achieved in these cases. To improve the approximation quality or convergence rate, one can use bases with higher orders. Naturally, the pairwise products of the bases from the linear elements span the space of quadratic polynomials. In this approach, the number of bases blows up quadratically with respect to the number of polygon vertices. Moreover, a number of pairwise products vanish along the polygonal boundaries, and thus do not contribute to inter-element continuity. Rand et al. provided a way of reducing the space of functions by piecing them together (Rand et al., 2014). The reduced space still forms a continuous polygonal spline space with quadratic reproducibility and retains inter-element continuity. The corresponding quadratic finite elements are called *generalized barycentric coordinates based quadratic serendipity elements (QSEs)*, which are a generalization of serendipity elements for quadrilateral cases (Zienkiewicz and Taylor, 2000; Arnold et al., 2002; Arnold and Awanou, 2011). Then, Sukumar presented the development of quadratic maximum-entropy serendipity shape functions on arbitrary planar polygons by maximizing an objective functional subject to the constraints for quadratic completeness (Rand et al., 2014). This method involves inexact and low efficient evaluations as it requires solving certain optimization problems numerically. Recently, Floater and Lai provided a method of constructing polygonal splines of order two or higher over a polygonal domain by using not only GBCs but also the local barycentric coordinates defined by triples of consecutive vertices of a polygon (Floater and Lai, 2016). Their construction for the quadratic case is simpler while slightly more time-consuming than the quadratic serendipity finite elements that appeared in Rand et al. (2014) (see statistics of running times in Section 5). In this paper, we develop algorithms for applications of partition-of-unity finite elements in functional/scattered data approximation, and while we use QSEs (Rand et al., 2014) as examples, our methods are applicable to other compactly supported finite elements on polygonal domains.

Consider a planar polygon Ω with n vertices $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ ordered counter-clockwise. Assume the polygon is not self-intersecting and has no more than three consecutive collinear vertices. Then each vertex \mathbf{v}_i can be associated with a GBC λ_i . Starting from a set of GBCs $\{\lambda_i\}_{i=1}^n$ defined by an n -sided polygon Ω , the pairwise products generate a quadratic function set $\{\mu_{i,j} := \lambda_i \lambda_j | i, j = 1, \dots, n\}$. Notice that $\mu_{i,j} = \mu_{j,i} := \lambda_i \lambda_j$, hence each vertex can be associated with a quadratic function $\mu_{i,i}$. And each edge or interior diagonal of the polygon can be associated with a function $2\mu_{i,j}$ with $i < j$. These quadratic bases naturally reproduce bivariate constant, linear and quadratic functions. To further reduce the basis number and guarantee the Lagrange property, these bases are transformed, using the linear combination method of Rand et al. (2014).

First, the coefficient 2 of each basis $2\mu_{i,j}$ is written as the sum of six numbers as

$$2 = c_{i,j}^{i,i} + c_{i,j}^{i,j} + c_{i,j}^{i,i-1} + c_{i,j}^{i,i+1} + c_{i,j}^{i,j-1} + c_{i,j}^{i,j+1}, \quad |i - j| > 1,$$

where:

$$\begin{aligned} c_{i,j}^{i,i} &= \frac{-2-2d_j}{2-d_i-d_j}, & c_{i,j}^{i,j} &= \frac{-2-2d_j}{2-d_i-d_j}, & c_{i,j}^{i,i-1} &= s \frac{\|\mathbf{l}_i \mathbf{v}_{i+1}\|}{\|\mathbf{v}_{i-1} \mathbf{v}_{i+1}\|}, \\ c_{i,j}^{i,i+1} &= s \frac{\|\mathbf{l}_i \mathbf{v}_{i-1}\|}{\|\mathbf{v}_{i-1} \mathbf{v}_{i+1}\|}, & c_{i,j}^{i,j-1} &= s \frac{\|\mathbf{l}_j \mathbf{v}_{j+1}\|}{\|\mathbf{v}_{j-1} \mathbf{v}_{j+1}\|}, & c_{i,j}^{i,j+1} &= s \frac{\|\mathbf{l}_j \mathbf{v}_{j-1}\|}{\|\mathbf{v}_{j-1} \mathbf{v}_{j+1}\|}, \end{aligned}$$

\mathbf{l}_i (resp. \mathbf{l}_j) is the intersection of segments $\mathbf{v}_i \mathbf{v}_j$ and $\mathbf{v}_{i-1} \mathbf{v}_{i+1}$ (resp. $\mathbf{v}_{j-1} \mathbf{v}_{j+1}$) (see Fig. 1 (right)),

$$d_i = 2 \frac{\|\mathbf{l}_i \mathbf{v}_{i,j}\|}{\|\mathbf{v}_i \mathbf{v}_j\|}, \quad d_j = 2 \frac{\|\mathbf{l}_j \mathbf{v}_{i,j}\|}{\|\mathbf{v}_i \mathbf{v}_j\|}, \quad s = \frac{2}{2 - d_i - d_j},$$

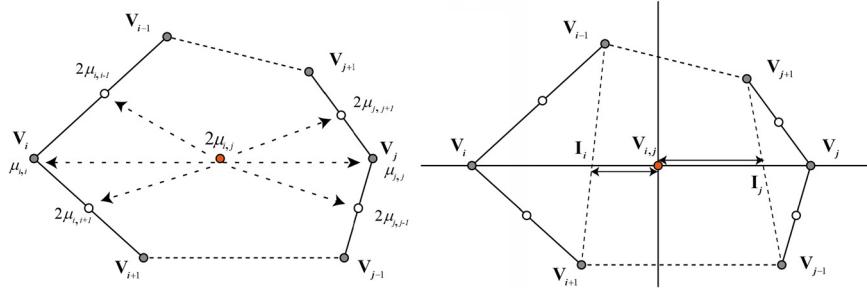


Fig. 1. Quadratic serendipity on a six-sided polygonal domain. The vertices and middle points of the edges of the polygonal domain are marked as filled and unfilled circles, respectively. (left) A basis $2\mu_{i,j}$ associating with an interior diagonal of the polygonal domain contributes to six different bases; (right) geometric meaning of the coefficients.

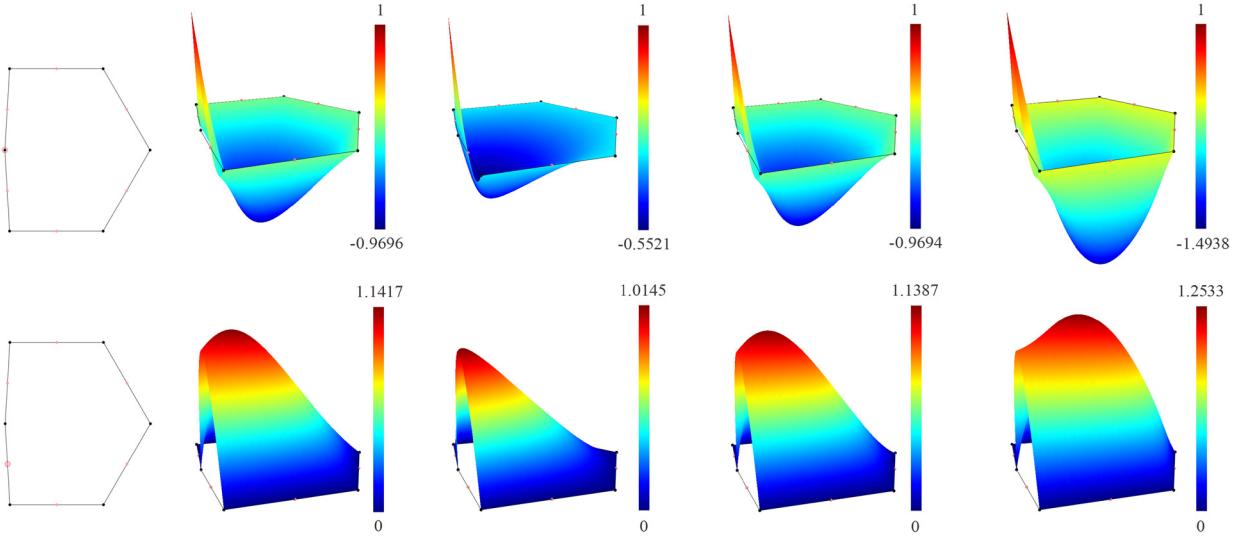


Fig. 2. Quadratic serendipity element bases on a six-sided polygonal domain. First column: polygonal domain, where the vertices and middle points of the edges of the polygonal domain are marked as black and pink points, respectively; the first and second rows are QSEs at a vertex and a middle point of an edge of the polygonal domain, respectively, where the corresponding vertex and the middle point are marked with circles in the first column. Second column to the last column: MVC based QSE bases, Wachspress coordinates based QSE bases, Poisson coordinates based QSE bases and natural neighbor coordinates based QSE bases. Values of the bases are color-coded, where colors at the top and bottom of the color bar represent the maximum and minimum values, respectively. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

and $\mathbf{v}_{i,j}$ is the middle point of the edge with end points \mathbf{v}_i and \mathbf{v}_j . Then, each basis $2\mu_{i,j}$ for $|i-j| > 1$ can be written as the sum of six terms, which only differ in their coefficients, as

$$2\mu_{i,j} = c_{i,j}^{i,i}\mu_{i,j} + c_{i,j}^{j,j}\mu_{i,j} + c_{i,j}^{i,i-1}\mu_{i,j} + c_{i,j}^{i,i+1}\mu_{i,j} + c_{i,j}^{j,j-1}\mu_{i,j} + c_{i,j}^{j,j+1}\mu_{i,j}, \quad |i-j| > 1.$$

The six terms on the right hand side of the above equation are reallocated to six different bases $\mu_{i,i}$, $2\mu_{i,i-1}$, $2\mu_{i,i+1}$, $\mu_{j,j}$, $2\mu_{j,j-1}$, $2\mu_{j,j+1}$ accordingly, which are associated with vertices \mathbf{v}_i , \mathbf{v}_j and their adjacent edges (see Fig. 1 (left)). More precisely, we obtain $2n$ bases on the polygonal domain Ω : $\xi_{i,i}$ and $\xi_{i,i+1}$ for $i = 1, \dots, n$, associated with vertex \mathbf{v}_i and the middle point of edge $\mathbf{v}_i\mathbf{v}_{i+1}$, respectively, where

$$\xi_{i,i} = \mu_{i,i} + \sum_{j,|i-j|>1} c_{i,j}^{i,i}\mu_{i,j} \quad \text{and} \quad \xi_{i,i+1} = 2\mu_{i,i+1} + \sum_{j,|i-j|>1} 2c_{i,j}^{i,i+1}\mu_{i,j} + \sum_{j,|i-j|>1} 2c_{i+1,j}^{i+1,j}\mu_{i+1,j}.$$

Finally, to guarantee the Lagrange property, we define

$$\psi_{i,i} = \xi_{i,i} - \xi_{i,i+1} - \xi_{i-1,i}, \quad \text{and} \quad \psi_{i,i+1} = 4\xi_{i,i+1}.$$

It has been proved in Rand et al. (2014) that the function set $\{\psi_{i,j} | i = j \text{ or } j = i + 1\}$ defined by the above method has the properties of constant, linear and quadratic precision and Lagrange like property. The polygonal domain with the function set $\{\psi_{i,j} | i = j \text{ or } j = i + 1\}$ forms a construction of quadratic serendipity elements on convex polygons derived from generalized barycentric coordinates. Examples of MVCs, Wachspress coordinates, Poisson coordinates, and natural neighbor coordinates based QSE bases on a six-sided polygonal domain are shown in Fig. 2, where basis functions corresponding to the vertex

Table 1

Statistics of the number of bases (denoted by Nbasis), M.E. and R.M.S.E. in each level of meshes and the ratios of M.E. and R.M.S.E. between successive levels of meshes in approximating Function 1 by using MVC based QSEs on VD, MVCs on VD and BCs on DT.

Ncell	MVC based QSEs on VD					MVCs on VD					BCs on DT				
	Nbasis	M.E.	Ratio	R.M.S.E	Ratio	Nbasis	M.E.	Ratio	R.M.S.E	Ratio	Nbasis	M.E.	Ratio	R.M.S.E	Ratio
16	83	1.95E-01		5.49E-02		34	7.33E-01		3.36E-01		11	8.92E-01		6.23E-01	
64	323	2.19E-02	8.91	5.19E-03	10.57	130	2.47E-01	2.97	8.33E-02	4.03	36	4.36E-01	2.04	2.01E-01	3.10
256	1283	2.76E-03	7.94	6.22E-04	8.35	514	7.58E-02	3.26	2.10E-02	3.96	145	1.64E-01	2.66	5.19E-02	3.87
1024	5123	3.36E-04	8.23	7.73E-05	8.05	2050	1.79E-02	4.23	5.20E-03	4.04	545	3.78E-02	4.33	1.27E-02	4.10
4096	20483	4.11E-05	8.16	9.41E-06	8.21	8197	3.98E-03	4.50	1.25E-03	4.17	2106	8.41E-03	4.50	3.09E-03	4.10

with a large interior angle are plotted to better distinguish the difference between each other. In the later sections, we will use the QSEs to fit discrete data and approximate analytic functions.

3. QSEs for functional data approximation

Quadratic serendipity elements are suitable for approximating given functions to any desired accuracy as long as the corresponding domains are appropriately discretized into polygonal elements. Moreover, the boundary values can be interpolated. QSEs are also adoptable for fitting discrete data when a data parameterization (i.e., a one-to-one mapping from the discrete data to a suitable domain, Floater and Hormann, 2005) is given. Owing to their quadratic polynomial reproduction property, QSEs can approximate functions with a quadratic convergence rate. In this section, we verify the quadratic convergence property of QSE bases in approximating an analytic function on a square domain. In addition, two greedy algorithms are proposed for adaptive QSEs approximation.

3.1. Notation

Assume that the input is a set of M vertices sampled from a given function $f(x, y)$ defined over domain $\Omega = [0, 1] \times [0, 1]$. We denote the sampling points on Ω as (α_i, β_i) , where $i = 1, \dots, M$. Assume that the domain Ω is divided into N polygonal cells with a tessellation $\mathcal{T} = \{\Omega_k \subset \Omega | k = 1, \dots, N\}$ and each polygonal cell Ω_k has L_k vertices ordered counter-clockwise and denoted by $\{\mathbf{v}_l | l = 1, \dots, L_k\}$. Then the fitting on each cell Ω_k is constructed as

$$Q_k(x, y) = \sum_{i=1}^{L_k} f(\mathbf{v}_{i,i}) \psi_{i,i} + f(\mathbf{v}_{i,i+1}) \psi_{i,i+1}, \quad (1)$$

where $\mathbf{v}_{i,i}$, $\mathbf{v}_{i,i+1}$, $\psi_{i,i}$ and $\psi_{i,i+1}$ are defined as described in the previous section. Then the approximate surface is a group of C^0 -connected patches $Q(x, y) = \sum_{k=1}^N Q_k(x, y)$. The fitting error at the parameter point (α_i, β_i) is defined as $e_i = |Q(\alpha_i, \beta_i) - f(\alpha_i, \beta_i)|$. Then the maximum error (M.E.) is $\max\{e_i | i = 1, \dots, M\}$ and the root-mean-square error (R.M.S.E.) is $\sqrt{\sum e_i^2/M}$.

3.2. Verification of the quadratic approximation property

Function 1.

$$f(x, y) = 0.1(1 + \cos(2\pi x) \sin(2\pi y)), \quad (x, y) \in \Omega = [0, 1] \times [0, 1].$$

In this section, we verify the quadratic approximation property of QSEs through experiments. The approximated analytic function we use here is defined as Function 1. The input is a grid mesh with $M = 513 \times 513$ vertices uniformly sampled from the function. The domain is progressively divided into $N = 16, 64, 256, 1024, 4096$ cells by a special Voronoi diagram generation method, called centroidal Voronoi tessellation (CVT) method (Liu et al., 2009), such that cells approximately have a uniform size (i.e., diameter) in the same mesh and are about half the size of the cells in successive meshes. Table 1 summarizes the number of bases, M.E., and R.M.S.E. in each level of meshes and the ratios of M.E. and R.M.S.E. between successive levels of meshes. As a comparison, we also report the fitting errors by using MVCs on the same Voronoi diagram (VD) and barycentric coordinates (BCs) on the Delaunay triangulation (DT) with the same number of faces. It can be seen that the expected quadratic convergence rate can be achieved using QSEs. Approximations using MVCs on VD and BCs on DT only show a linear rate of convergence. It is also worth noting that the number of bases of QSEs on VD only increases roughly linearly with the number of cells. And the number of QSE bases is less than three times of the number of MVC bases on VD and less than ten times of the number of BC bases on DT. The resultant surfaces constructed on 64 cells are shown in Fig. 3.

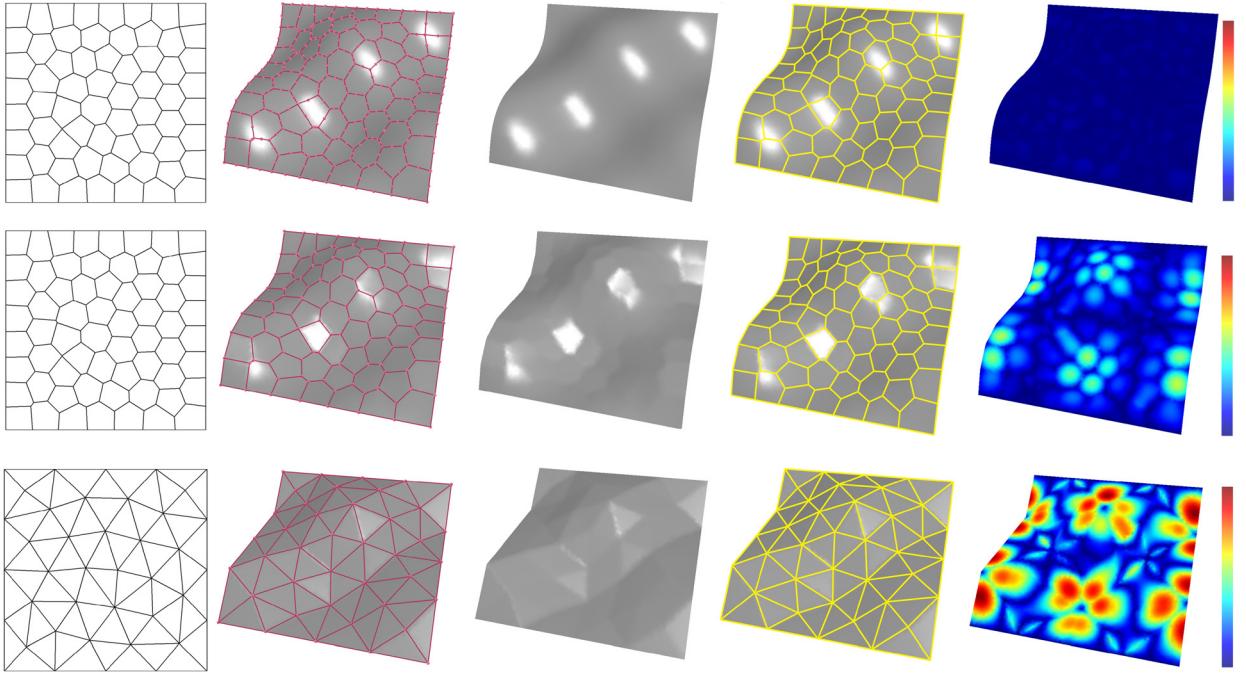


Fig. 3. Approximation results of Function 1 by using 64 elements. Rows from top to bottom are approximation results of MVC based QSEs on VD, MVCs on VD, and BCs on DT. Columns from left to right are tessellations with 64 elements, interpolated points (red dot points), result surfaces, surfaces with curved cell boundaries and color-coded fitting errors by the same color bar (dark red indicates the largest error of the three cases and blue indicates the value of 0).

3.3. Voronoi diagram based QSE fitting/approximation algorithms

Note that, if we adopt uniformly partitioned domains for surface fitting, such as uniformly refined polygonal meshes in Lai and Slavov (2016), a large number of uniform cells may be needed to reach a pre-specified fitting error threshold, especially when the approximated function oscillates unevenly. To reduce the cell number and minimize the fitting error in the approximation, an adaptive cell generation method is need. In this section, we propose two algorithms for approximating analytic functions or discrete data (with a given parametrization) using MVC based QSEs. Both algorithms take a rectangular gridded height field as input, and approximate it with a group of C^0 -connected QSEs constructed on a Voronoi mesh, i.e., a network of Voronoi cells tessellating the function domain. We begin with a coarse Voronoi mesh that is restricted to the function domain. For example, the initial mesh used here is the restricted Voronoi diagram generated by the four corners of the domain. Then on each iteration, the Voronoi mesh is refined by using a greedy point insertion strategy, which seeks to insert one or several points in the domain to locally lower the approximation error while keeping the previously inserted points fixed. By refining the Voronoi cells, the approximation quality of the associated QSEs is improved progressively. The process of refinement continues until the approximation algorithm reaches a desired error threshold or exhausts a pre-defined budget on the number of elements.

In each iteration, let p_i be the parametric point of the point $v_i \in \mathbb{R}^3$ with the highest error in the input data and T be the dual Delaunay triangulation of the current Voronoi mesh. Then, two different Voronoi mesh refinement methods are given as follows:

The first method is based on eliminating the Delaunay triangle f_j of T where p_i is located. A point is inserted to the circumcenter of f_j if it is inside the domain, otherwise, the middle point of the longest edge of f_j is inserted. Pseudo-code is given in Algorithm 1. Our refinement method is similar to the Delaunay refinement approach (Shewchuk, 2002). Although our algorithm inevitably introduces cells with short edges or large angles, as can be observed in Fig. 6–7, which may violate the criteria for a “well-shaped polygonal element” (Gillette et al., 2012; Sieger et al., 2010), experiments show that it usually generates polygonal meshes that are good enough for the functional data approximation applications.

The second method is based on progressively eliminating the highest approximation error. It constructs Voronoi cells with a vertex coincident with p_i , such that the associated QSEs can interpolate the point v_i with the currently highest error. Pseudo-code is given in Algorithm 2. Specifically, a circle C centered at p_i continues to grow until it touches a vertex t_j of the dual Delaunay triangulation T . Then two points t_1^* and t_2^* are sampled on this circle. Furthermore, positions of t_1^* and t_2^* restricted on the circle C are optimized by the Lloyd’s method (Lloyd, 1982), such that the existing points and newly added points distribute evenly (see Fig. 4). Voronoi cells associated with t_j , t_1^* and t_2^* have p_i as a vertex, therefore, the associated QSEs interpolate the data point v_i associated with parametric point p_i .

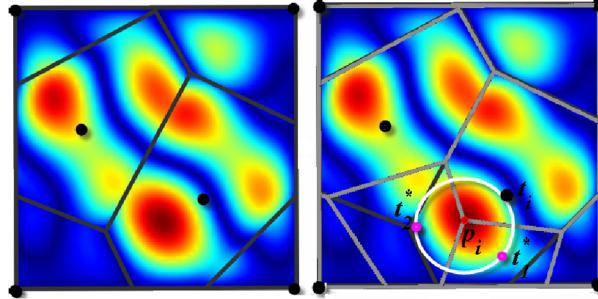


Fig. 4. The second Voronoi mesh refinement approach. (left) Six points (black dots) and their corresponding Voronoi cells (black polygons). Warm colors represent high errors. (right) A circle (white) is centered at the parametric point p_i with the highest error and touches a previously inserted point t_i . Two points (pink) on the circle are added to the point set, and p_i becomes a vertex of the updated Voronoi mesh (gray polygons).

Algorithm 1: Voronoi mesh refinement algorithm by inserting the circumcenter of the Delaunay triangle with the highest error.

```

Input : a Voronoi tessellation  $\mathcal{T}$  of the function domain  $\Omega$  and fitting errors  $\{e_i | i = 1, \dots, M\}$ 
Output: a refined Voronoi tessellation  $\mathcal{T}'$  of the function domain  $\Omega$ 
1 let  $T$  be the dual Delaunay triangulation of the Voronoi tessellation  $\mathcal{T}$ ;
2 find the parametric point  $p_i$  with the highest fitting error;
3 collect all the triangles in  $T$  that contain  $p_i$ , and denote the triangle set by  $F = \{f_{i_j} | j = 1, \dots, K\}$ ;
4 choose a triangle in  $F$  with the highest R.M.S.E., and denote it by  $f_j$ ;
5 compute the circumcenter of  $f_j$ , and denote it by  $c_j$ ;
6 if  $c_j \in \Omega$  then
7   | insert  $c_j$  into the Voronoi tessellation  $\mathcal{T}$  ;
8 else
9   | insert the middle point of the longest edge of  $f_j$ ;
10 end
11 return the new Voronoi tessellation  $\mathcal{T}'$ ;

```

Algorithm 2: Voronoi mesh refinement algorithm by refining the facet with the highest data approximation error.

```

Input : a Voronoi tessellation  $\mathcal{T}$  of the function domain  $\Omega$  and fitting errors  $\{e_i | i = 1, \dots, M\}$ 
Output: a refined Voronoi tessellation  $\mathcal{T}'$  of the function domain  $\Omega$ 
1 find the parametric point  $p_i$  with the highest fitting error;
2 find the nearest site of the Voronoi tessellation  $\mathcal{T}$ , and denote it by  $t_j$ ;
3 construct a circle  $C$  that is centered at  $p_i$  and has a radius of  $|t_j - p_i|$ ;
4 randomly sample two points on the circle  $C$ , denoted by  $t_1^*$  and  $t_2^*$ , and add them to the Voronoi tessellation  $\mathcal{T}$ ;
5 repeat
6   move  $t_1^*$  and  $t_2^*$  to the centroids of their corresponding Voronoi regions; /* Lloyd's iteration */
7   project  $t_1^*$  and  $t_2^*$  onto the circle  $C$ ;
8   update the Voronoi tessellation;
9 until iteration number > 5;
10 return the new Voronoi tessellation  $\mathcal{T}'$ ;

```

As the two approximation algorithms differ only in the Voronoi mesh refinement approaches, we also refer to the two greedy algorithms Algorithm 1 and Algorithm 2 as the same as their refinement approaches, when there is no ambiguity. In our experiments, we find that while both algorithms generate similar results in the sense that they reach a similar error threshold by using almost the same number of cells, an advantage of the second algorithm is that it inserts two points in each iteration, so reaches the error threshold by using fewer iterations. For brevity, we only report the results generated by Algorithm 1 here. We also present a comparison of approximation abilities of MVC based QSEs/MVCs on VD and BCs on DT through experiments, where all the cells are generated by Algorithm 1. Here, we only use MVC based QSEs to illustrate the feasibility of the proposed greedy algorithms. Comparisons between different coordinates based QSEs will be reported in the following section.

Function 2.

$$f(x, y) = 10^{-5} \exp^{7((x-\frac{1}{4})^2 + (y-\frac{1}{3})^2)}, \quad (x, y) \in \Omega = [0, 1] \times [0, 1].$$

Two models are used as test data sets. The first data set is uniformly sampled from the analytic Function 2. The plots of R.M.S.E. against the cell number in Fig. 5 clearly show that the increasing cell numbers decrease the approximation error. In particular, the algorithm with MVC based QSEs on VD uses far fewer cells (equally, iteration steps) to reach the same error threshold than with other two kinds of finite elements. The approximation results with R.M.S.E. just falling below

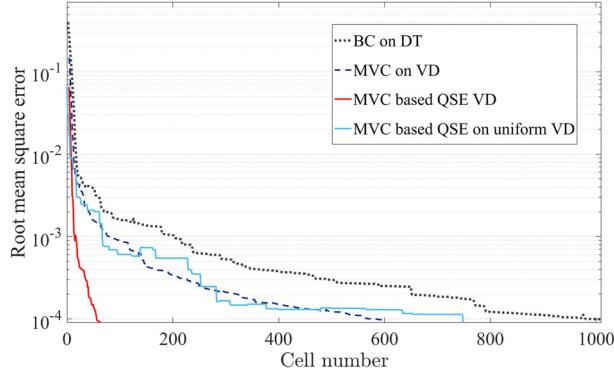


Fig. 5. Plots of R.M.S.E. (log-scale) against the cell number for the approximation of Function 2 by using Algorithm 1, where three elements including MVC based QSEs/MVCs on VD and BCs on DT are used. The cell number is slightly greater than the iteration number for the three kind of finite elements. For example, cell number is equal to iteration number plus one for the case of BC on DT. To illustrate the efficacy of the adaptive schemes, we also include by way of reference, the approximation result by using MVC based QSEs on uniform Voronoi cells generated by the farthest point sampling method.

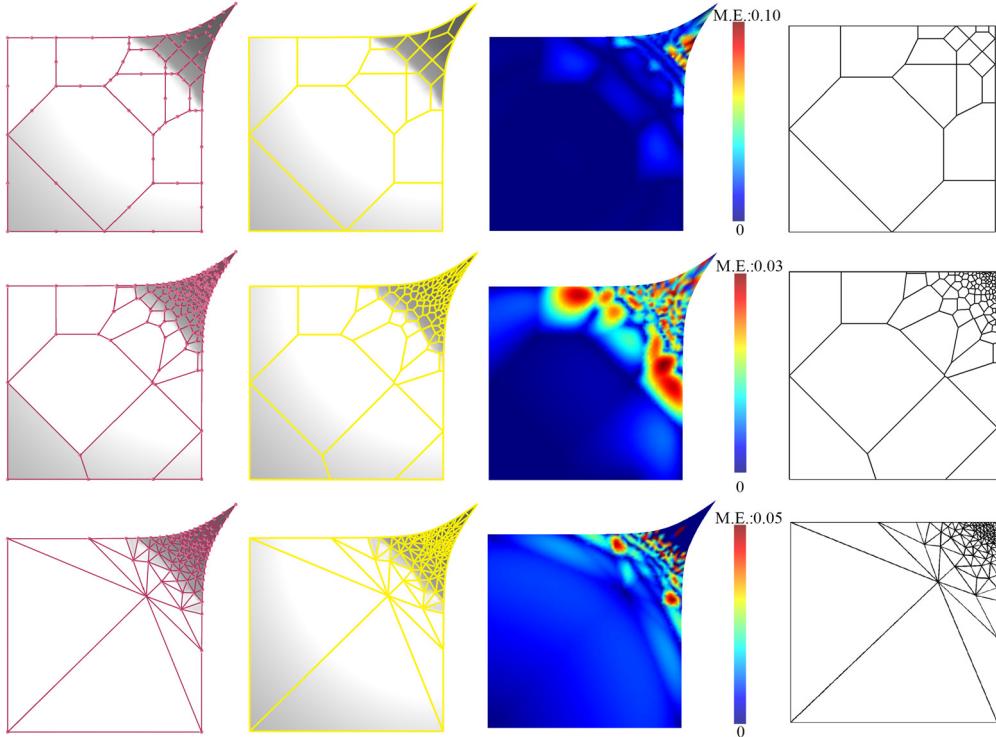


Fig. 6. Approximation results of Function 2 by using MVC based QSEs on VD with 19 cells (first row), MVCs on VD with 116 cells (second row), and BCs on DT with 156 faces (third row). Columns from left to right are the result surfaces with interpolated points, result surfaces with curved cell boundaries, color-coded approximation errors, and the tessellations generated by Algorithm 1. The R.M.S.E threshold is 0.01.

the threshold of 0.01 are shown in Fig. 6, where the first, second and third rows are the approximation results by MVC based QSEs on VD (19 cells), MVCs on VD (116 cells), and BCs on DT (156 triangular faces). The first and second columns show the result surfaces with interpolated points and curved cell boundaries, while the third column shows the color-coded approximation errors and the last column shows the tessellations generated by Algorithm 1.

The algorithm can be naturally applied to image data or scattered data points, as long as there is an appropriate parametrization. Fig. 7 shows approximation results with a R.M.S.E threshold of 0.001 on the second test model, i.e., a triangular mesh with its 2D parametrization. Similar to the case of analytic function above, MVC based QSEs use far fewer finite elements (equally, iterations) than the other two types of finite element bases to reach the same error threshold. Furthermore, MVC based QSEs generate a visually smoother surface.

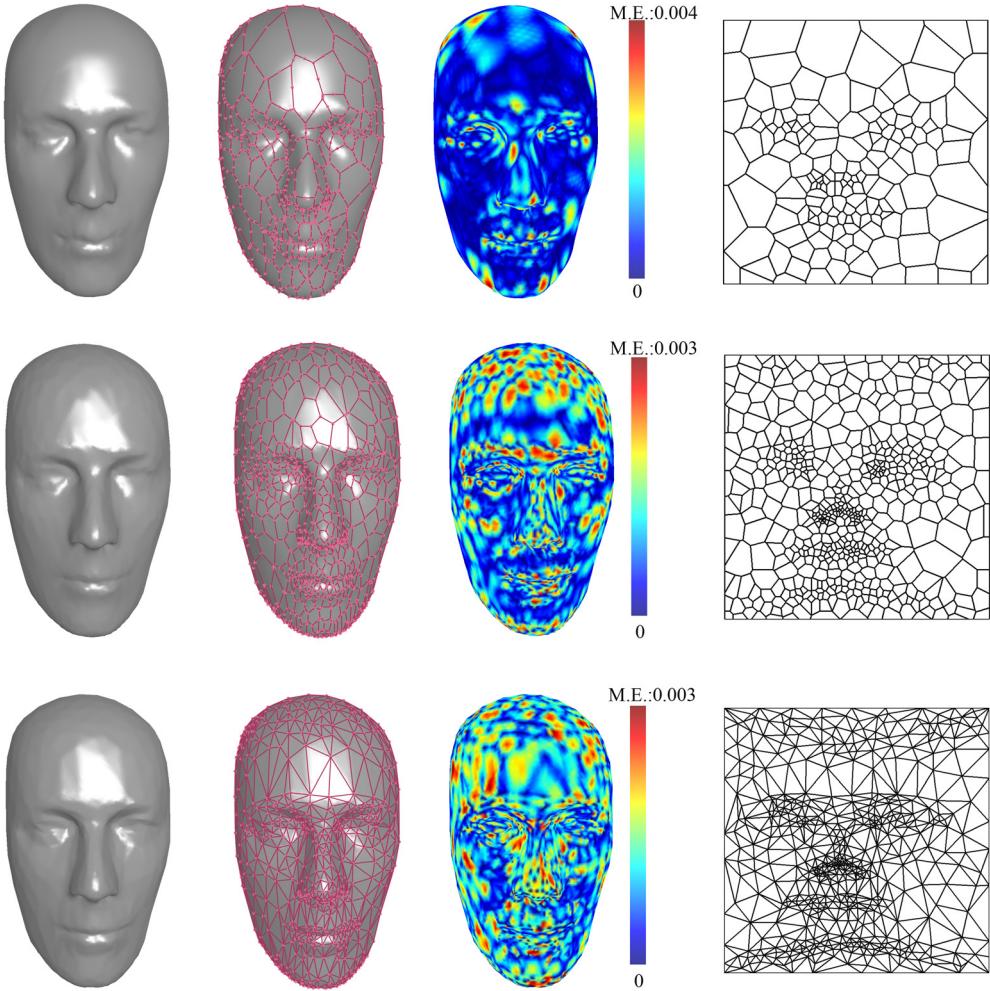


Fig. 7. Approximation results of face model (with 8k vertices) by using MVC based QSEs on VD with 156 cells (first row), MVCs on VD with 541 cells (second row), and BCs on DT 1054 cells (third row). Columns from left to right are the result surfaces, result surfaces with interpolated points, color-coded approximation errors, and the tessellations generated by Algorithm 1. The R.M.S.E threshold is 0.001.

4. Optimized Voronoi mesh generation

As can be observed from Fig. 6 and Fig. 7 that input data with more sharp features may need more cells for the approximation. However, there is no guarantee that Algorithm 1 can recover all these features, as for example, in the mouth region in Fig. 7, see also the approximation around various features in Fig. 9 and Fig. 10. One possible reason is that the feature lines on the parametric domain may cross Voronoi cells, hence the features are blurred. Intuitively, cells aligning along the feature lines on the domain can be helpful for recovering the sharp features, as the QSE surface is only C^0 on cell boundaries while it is smooth in the interior of a Voronoi cell. In this section, we take Voronoi meshes generated by Algorithm 1 as input and further optimize the Voronoi meshes by using an L_2 optimization method toward feature-conforming Voronoi meshes, such that the approximation results of input image data with sharp features can be further improved.

4.1. Objective function

Assume that $f(\mathbf{x})$ is a function defined over a compact 2D domain Ω , and the domain is discretized by a Voronoi tessellation $\mathcal{T} = \{\Omega_k \subset \Omega | k = 1, \dots, N\}$ of a site set $X = \{\mathbf{x}_k\}_{k=1}^N$, where Ω_k is a Voronoi cell of site $\mathbf{x}_k \in X$. We can define the approximation error in L_2 sense as follows:

$$E(\{Q_k\}_{k=1}^N, X) = \sum_{k=1}^N \int_{\Omega_k} |f(\mathbf{x}) - Q_k(\mathbf{x})|^2 d\mathbf{x}, \quad (2)$$

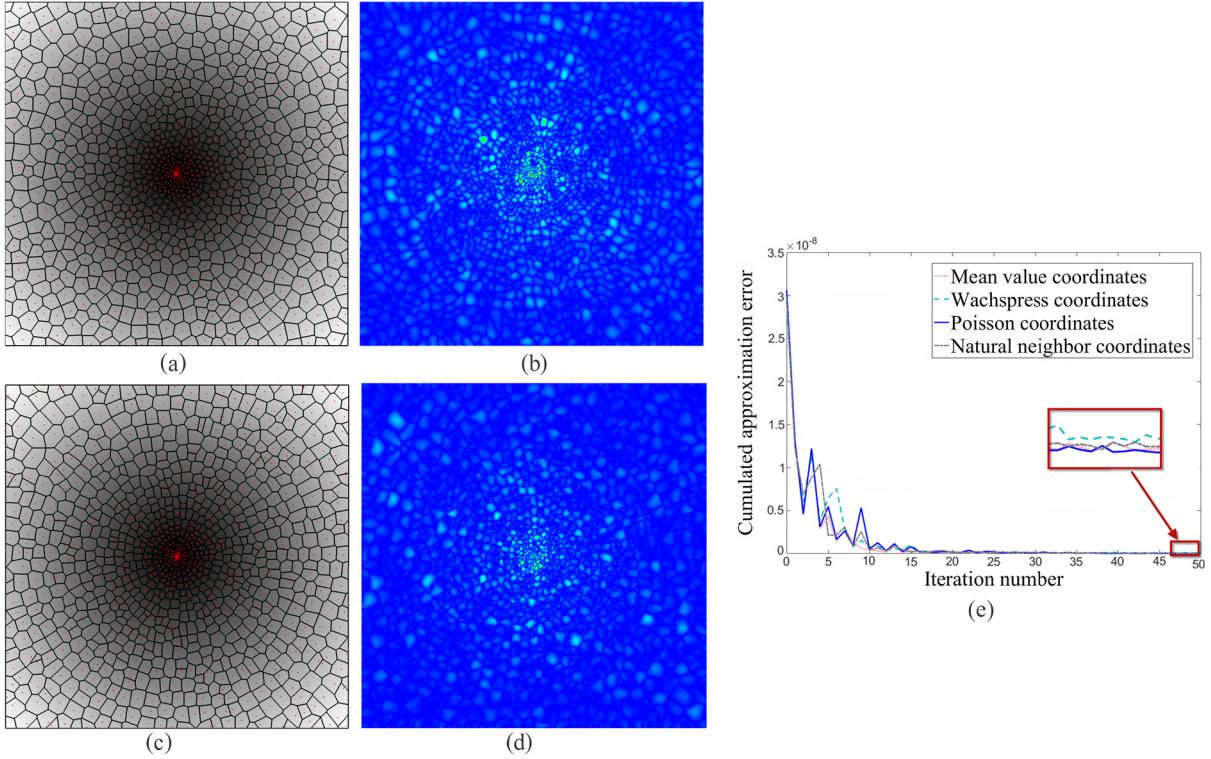


Fig. 8. Approximation results of function $f(x, y) = \sqrt{x^2 + y^2}$, $(x, y) \in [-1, 1] \times [-1, 1]$ by using QSEs on optimized Voronoi diagram with 1000 cells. (a) Initial Voronoi diagram (where the domain is color-coded by the value of the given function); (b) color-coded approximation error on the initial mesh (M.E.: $6.9386E - 05$, R.M.S.E: $9.9629E - 07$); (c) Voronoi diagram optimized after 50 iterations; (d) color-coded approximation error on the optimized mesh (M.E.: $3.4799E - 05$, R.M.S.E: $4.4063E - 07$); (e) cumulated approximation errors of the results (by using four different generalized barycentric coordinates based QSEs) vs. fitting iteration numbers.

where $Q_k = \sum_{i=1}^{L_k} f(\mathbf{v}_{i,i})\psi_{i,i} + f(\mathbf{v}_{i,i+1})\psi_{i,i+1}$ is the approximate function on Ω_k as defined in Equation (1). Note that the above L_2 approximation error only depends on the site positions X , as the bases $\psi_{i,i}$ or $\psi_{i,i+1}$ as well as their coefficients are defined by the positions of vertices of the cells (equally, the sites). An efficient method for minimizing Equation (2) usually requires the computation of gradient. However, the bases are rational functions with respect to the positions of vertices of cells, and the expressions of vertex positions with respect to site positions also involve multiple square root terms. Hence, the direct computation of gradient is complicated and time-consuming. To simplify the computation, we minimize a slightly modified objective function as follows:

$$E(\{Q_k^*\}_{k=1}^N, X) = \sum_{k=1}^N \int_{\Omega_k} |f(\mathbf{x}) - Q_k^*(\mathbf{x})|^2 d\mathbf{x}, \quad (3)$$

where $Q_k^* = \sum_{i=1}^{L_k} \alpha_{i,i}^k \psi_{i,i} + \alpha_{i,i+1}^k \psi_{i,i+1}$ is the optimal approximation on Ω_k , $\alpha_{i,i}^k$ and $\alpha_{i,i+1}^k$ are the coefficients associated with bases $\psi_{i,i}$ and $\psi_{i,i+1}$ on Ω_k , respectively. Intuitively, the modified objective function allows the coefficients $\alpha_{i,i}$ (or $\alpha_{i,i+1}$) associated with the same vertex (or midpoint of an edge) to be different on the adjacent cells, hence, relaxes the C^0 continuity between adjacent patches. This modification is reasonable since if the approximation surface is close enough to the given continuous function, then the different coefficients associated with the same vertex will be close to each other. A further stitching operation can be performed to achieve C^0 continuity. This patch-stitching procedure leads to an increase in approximation errors. To avoid the deterioration in approximation quality, an intuitive way of finding the best stitched patches is to solve for the coefficients in a linear least square sense. Specifically, based on the tessellation of the minimizer of Equation (3), we solve a linear least squares problem in the entire domain by restricting all the coefficients associated with the same vertex/midpoint of an edge to be equal. A simpler, yet more efficient method adopted in this paper for ensuring the C^0 continuous between patches is to replacing the coefficients associated with the same vertex with the same value, e.g. the average of all the associated coefficients. It is because that Voronoi meshes with a large number of cells (relative to image resolution) generate unstitched patches with very small gaps between them. In other words, the coefficients of the bases corresponding to the same vertices/midpoints, though not equal, are very close to each other. Moreover, averaging

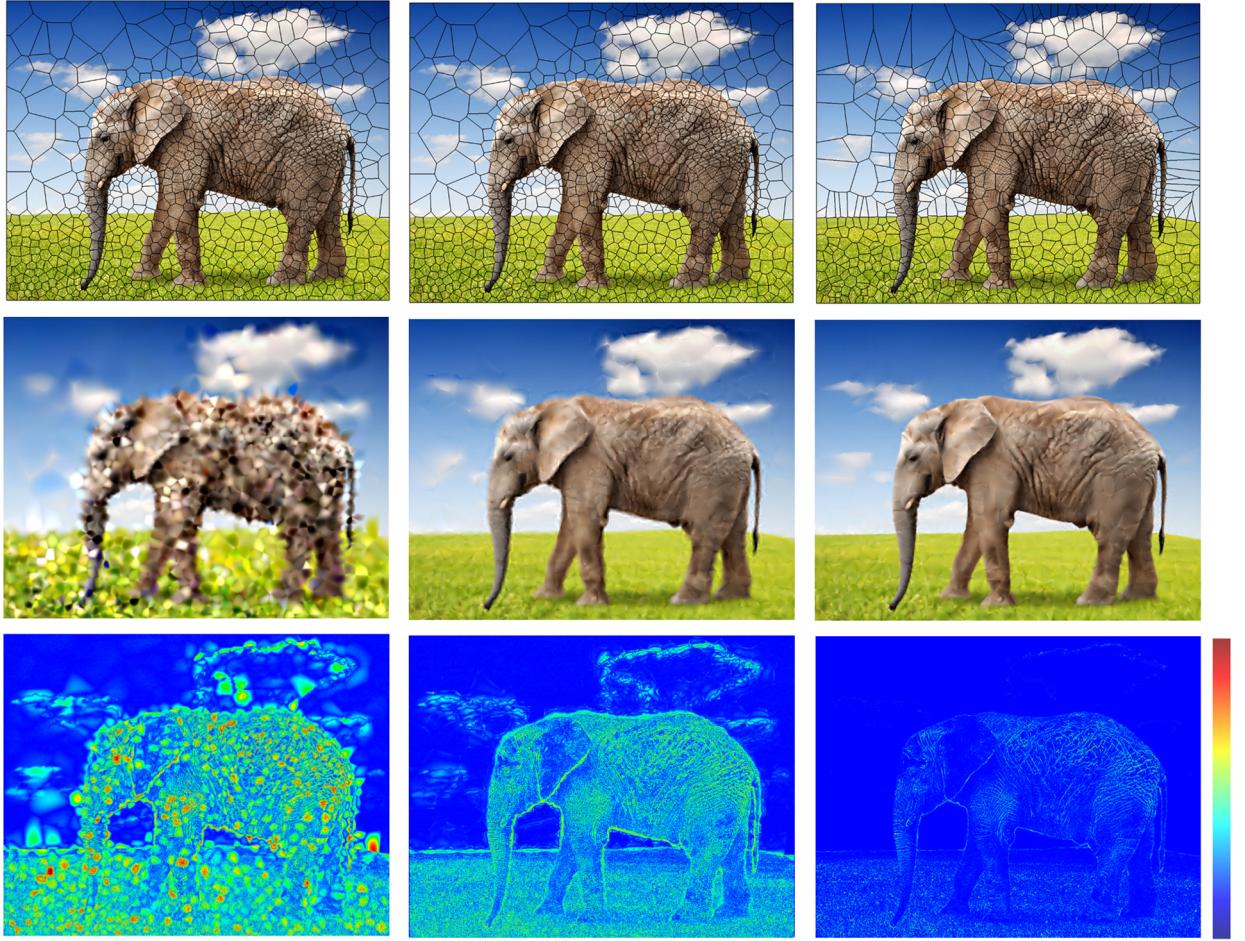


Fig. 9. Approximation on Elephant by using Poisson coordinates based QSEs on an optimized Voronoi diagram. Rows from top to bottom are: Voronoi diagrams with 1000 cells, approximation images and color-coded approximation errors. The first ($M.E : 2.2387$, $R.M.S.E. : 0.2887$) and second columns ($M.E : 1.1684$, $R.M.S.E. : 0.0872$) are the approximation results on the initial Voronoi diagram with and without stitching, respectively. The last column ($M.E : 0.9985$, $R.M.S.E. : 0.0813$) is the approximation results after 30 iterations of the L_2 optimization procedure without patches stitched for C^0 continuity. The same color code is used in the error visualization in all columns, where colors at the top and bottom of the color bar correspond to values of 1.17 and 0, respectively.

these coefficients have no significant influence, either on the approximation error or on the visual effect. Whereas, if the given function is discontinuous (such as image data), coefficients associated with the same vertex may be required to be different, especially on discontinuous regions. We will show this in Section 5. In this case, the stitching step is unnecessary.

4.2. Optimization method

To simplify the computation, we optimize alternately between the optimal approximations $\{Q_k^*\}_{k=1}^N$ and site set X . For a fixed X (equally, partition \mathcal{T}), the optimal approximation $Q_i^*(\mathbf{x})$ on each cell Ω_k , i.e. the coefficients $\alpha_{i,i}^k$ and $\alpha_{i,i+1}^k$, can be obtained by solving a least-squares problem on each cell individually. On the other hand, with the fixed functions $\{Q_k^*\}_{k=1}^N$, the gradient of the objective function in Equation (3) can be derived as follows :

$$\frac{\partial E(X)}{\partial \mathbf{x}_i} = \sum_{j \in J_i} \int_{\Omega_{i,j}} (|f(\mathbf{x}) - Q_i^*(\mathbf{x})|^2 - |f(\mathbf{x}) - Q_j^*(\mathbf{x})|^2) \frac{\mathbf{x} - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} ds,$$

where J_i is the set of indices of the sites whose Voronoi cells are adjacent to Ω_i . Then we adopt a modified gradient descent method proposed in Chen et al. (2014) for an efficient solution to the L_2 optimization. For the sake of completeness, we give a brief description of the L_2 optimization method here. Beginning with an initialization, we optimize all the positions of the sites by:

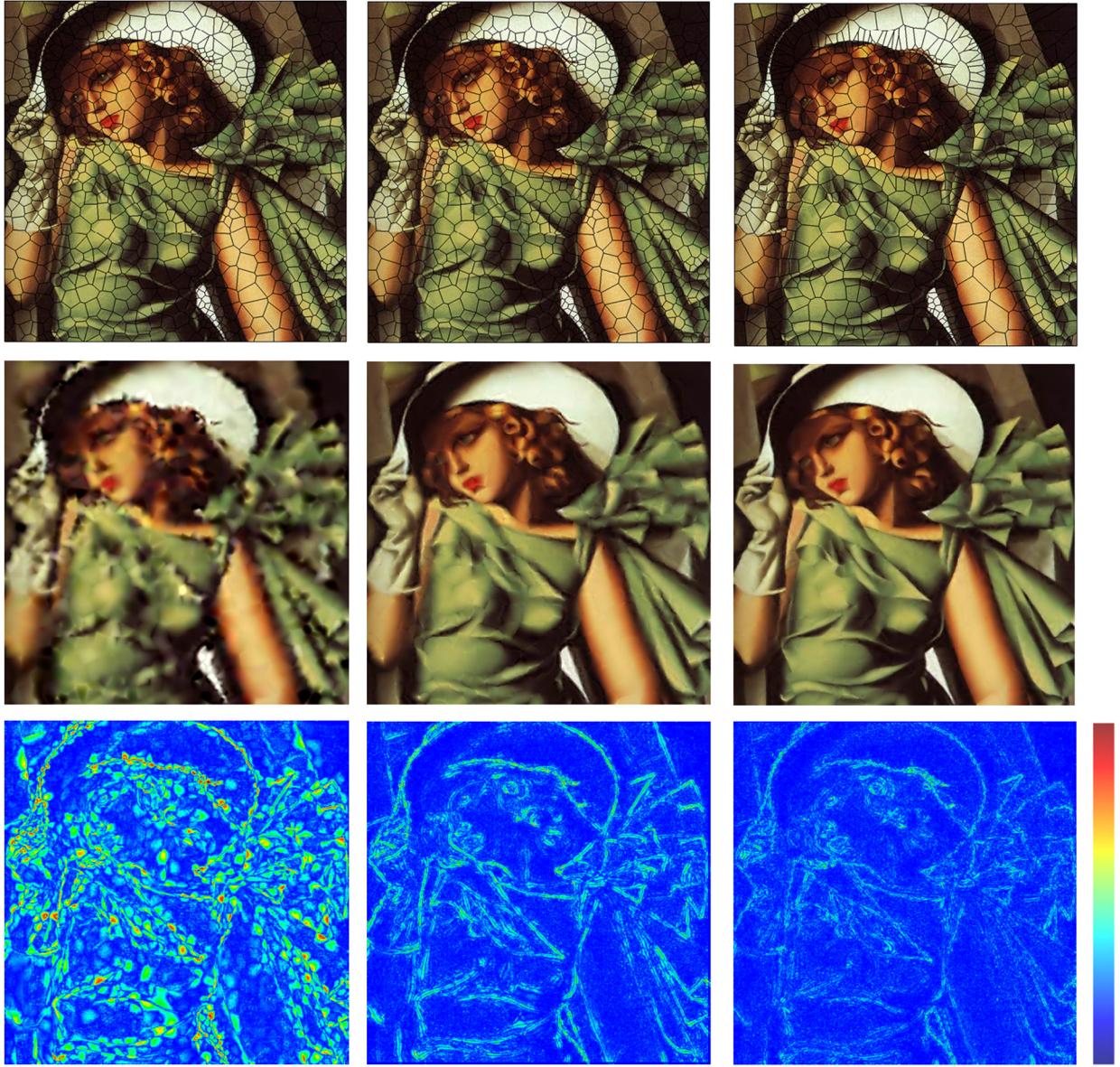


Fig. 10. Approximation on Lady by using Poisson coordinates based QSEs on an optimized Voronoi diagram. Rows from top to bottom are: Voronoi diagrams with 1000 cells, approximation images and color-coded approximation errors. The first ($M.E : 1.6866$, $R.M.S.E. : 0.1553$) and second columns ($M.E : 0.8977$, $R.M.S.E. : 0.0438$) are the approximation results on the initial Voronoi diagram with and without stitching, respectively. The last column ($M.E : 0.8984$, $R.M.S.E. : 0.0306$) shows the approximation results after 30 iterations of the L_2 optimization procedure without stitching. The same color code is used in the error visualization in all columns, where colors at the top and bottom of the color bar correspond to values of 0.85 and 0, respectively.

$$\mathbf{x}_i^{(j+1)} = \mathbf{x}_i^{(j)} - \delta_i^{(j)} \frac{\partial E(X)}{\partial \mathbf{x}_i} / \left| \frac{\partial E(X)}{\partial \mathbf{x}_i} \right|, i = 1, \dots, N,$$

where j is the index of the current iteration, and $\delta_i^{(j)}$ is the step length for \mathbf{x}_i . The step length $\delta_i^{(j)}$ is controlled by the formula

$$\delta_i^{(j)} = \delta_i^{(0)} \left(\frac{1}{2} \right)^{\frac{j}{I_{\max}-j}}, i = 1, \dots, N,$$

where I_{\max} is a specified maximum number of iterations and $\delta_i^{(0)}$ is the initial step length for site \mathbf{x}_i , which is set as the square root of the area of Ω_i .

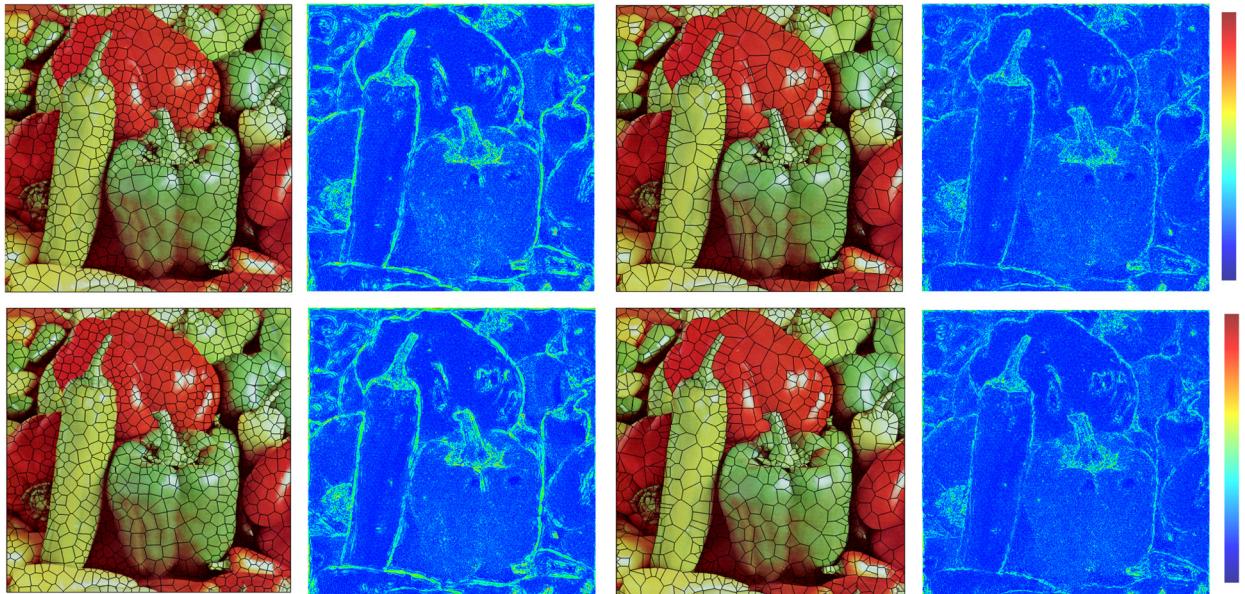


Fig. 11. Comparison of approximations using Wachspress coordinates based QSEs constructed by two different methods (Floater and Lai, 2016; Rand et al., 2014). Input image resolution: 512×512 . Rows on the top and at the bottom are the approximation results by QSEs constructed by Floater and Lai (2016) and Rand et al. (2014), respectively. The first and third columns are the initial and final (after 100 optimization steps) Voronoi diagrams with 1000 cells. And the second and fourth columns are the color coded approximation errors for the initial approximation results ($M.E : 0.7479$, $R.M.S.E. : 0.0768$ for Floater and Lai, 2016, and $M.E : 0.7964$, $R.M.S.E. : 0.0785$ for Rand et al., 2014) and the final results ($M.E : 0.6866$, $R.M.S.E. : 0.0588$ for Floater and Lai, 2016, and $M.E : 0.6902$, $R.M.S.E. : 0.0584$ for Rand et al., 2014). The same color code is used in the error visualization in all columns, where colors at the top and bottom of the color bar correspond to values of 0.70 and 0, respectively. Running times for our approximation algorithm using QSE constructions from Floater and Lai (2016) and Rand et al. (2014) are 805s and 717s, respectively.

5. Experimental results

This section presents some examples to demonstrate the optimized approximation capability of our algorithms. First, we approximate an analytic function $f(x, y) = \sqrt{x^2 + y^2}$, where $(x, y) \in [-1, 1] \times [-1, 1]$, by using four different barycentric coordinates (including mean value coordinates, Wachspress coordinates, Poisson coordinates and natural neighbor coordinates) based QSEs. Totally 1000 sites are generated in the initial step and then the Voronoi diagrams are optimized by the method proposed in Section 4. The plots of the cumulated errors of the approximate surfaces against the iteration number are shown in the right-hand-most figure in Fig. 8, from which we can see that the optimization converges after about 30 iterations and the Poisson coordinates based QSEs perform slightly better than other coordinates in the sense of reaching a lower cumulated approximation error. The initial Voronoi diagram and the Voronoi diagram after 50 iterations of the optimization procedure, and the color-coded approximation errors of the corresponding approximation results by using Poisson coordinates based QSEs are also shown in Fig. 8, where the patches on the Voronoi cells are stitched to become C^0 continuous by simply replacing the coefficients associated with the same vertex with the same value, i.e., the average of all associated coefficients. It can be observed that, the Voronoi cells tend to be radially arranged and concentrated in the center of the domain after optimization. We also apply our algorithm to image approximation and two examples are shown in Fig. 9 and Fig. 10, respectively. Rows from top to bottom are Voronoi diagrams with 1000 cells, approximation images on the Voronoi diagram, and color-coded approximation errors. Again, Poisson coordinates based QSEs are adopted since they perform slightly better than the others. The stitched and unstitched approximation results on initial Voronoi diagrams instead of the final Voronoi diagrams are shown in the first and second columns, respectively, to better display the artifacts introduced by the stitching, as the visual difference between the stitched and unstitched results are less obvious on the final Voronoi diagrams. It can be seen that the approximation results without stitching are visually more pleasant than the stitched ones. The approximation images after 30 iterations of the optimization procedure without stitching are shown in the last columns in both Fig. 9 and Fig. 10. We can observe that cells tend to align along the feature lines of the images, and the errors are lowered and features are better preserved after optimization. Our algorithm can also be adapted to other quadratic serendipity finite elements, such as the elements proposed in Floater and Lai (2016). Through experiments we find that our algorithm generates results with similar approximation errors using quadratic serendipity finite elements proposed in Rand et al. (2014) and Floater and Lai (2016). Approximation results of using Wachspress based QSEs constructed by Floater and Lai (2016) and Rand et al. (2014) are shown in Fig. 11, where the former method (804s) took up slightly more time than the later method (717s). We believe that our algorithm can also be easily adapted to polygonal elements with higher orders (e.g., Floater and Lai, 2016).

6. Conclusion

In this paper, we construct piecewise approximations of functional data on arbitrary 2D bounded domains, using generalized barycentric, quadratic serendipity elements (QSEs) on Voronoi cells, and compare them to barycentric finite elements on triangulations. We compare approximation qualities of linear and quadratic finite elements through several numerical experiments, using different kinds of barycentric and generalized barycentric finite elements. Two adaptive mesh refinement methods and an optimization method are presented for data approximation by using QSEs. Experimental results show space/accuracy advantages for these finite elements on polygonal domains against traditional finite elements over simplicial meshes.

Two non-uniform Voronoi mesh generation algorithms are also presented and specifically designed for the purpose of surface fitting. They help achieve good tradeoffs between approximation accuracy with respect to the number of Voronoi cells. We shall explore such tradeoffs in future work on isogeometric analysis, using quadratic serendipity finite elements based on generalized barycentric coordinates over polygonal domains.

Acknowledgements

The research of Juan Cao and Zhonggui Chen was supported by the National Natural Science Foundation of China (No. 61472332, 61572020), and the Natural Science Foundation of Fujian Province of China (No. 2018J01104). The research of Chandrajit Bajaj was supported in part by NIH grants R01GM117594, R41GM116300 and Dell-Seton 201602388.

References

- Adams, M.D., 2011. A flexible content-adaptive mesh-generation strategy for image representation. *IEEE Trans. Image Process.* 20, 2414–2427.
- Adams, M.D., 2013. A highly-effective incremental/decremental Delaunay mesh-generation strategy for image representation. *Signal Process.* 93, 749–764.
- Amirfakhrian, M., 2010. Best approximation of multivariate functions in l_1 and l_2 by optimization. *Math. Sci.* 4, 205–219.
- Arnold, D.N., Awanou, G., 2011. The serendipity family of finite elements. *Found. Comput. Math.* 11, 337–344.
- Arnold, D.N., Boffi, D., Falk, R.S., 2002. Approximation by quadrilateral finite elements. *Math. Comput.* 71, 909–922.
- Cao, J., Li, X., Chen, Z., Qin, H., 2012. Spherical DCB-spline surfaces with hierarchical and adaptive knot insertion. *IEEE Trans. Vis. Comput. Graph.* 18, 1290–1303.
- Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R., 2001. Reconstruction and representation of 3D objects with radial basis functions. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 67–76.
- Chen, Z., Xiao, Y., Cao, J., 2014. Approximation by piecewise polynomials on Voronoi tessellation. In: Geometric Modeling and Processing 2014. Graph. Models 76, 522–531.
- Dahmen, W., Micchelli, C.A., Seidel, H.-P., 1992. Blossoming begets B-spline bases built better by B-patches. *Math. Comput.* 59, 97–115.
- Dunkl, C.F., 1987. Orthogonal polynomials on the hexagon. *SIAM J. Appl. Math.* 47, 343–351.
- Farouki, R.T., Goodman, T.N., Sauer, T., 2003. Construction of orthogonal bases for polynomials in Bernstein form on triangular and simplex domains. *Comput. Aided Geom. Des.* 20, 209–230.
- Fedele, G., Ferrise, A., 2011. Explicit solution of the finite time l_2 -norm polynomial approximation problem. *Appl. Comput. Math.* 217, 8354–8359.
- Floater, M.S., 2015. Generalized barycentric coordinates and applications. *Acta Numer.* 24, 161–214.
- Floater, M.S., Hormann, K., 2005. Surface parameterization: a tutorial and survey. In: Advances in Multiresolution for Geometric Modelling. Springer, pp. 157–186.
- Floater, M.S., Lai, M.-J., 2016. Polygonal spline spaces and the numerical solution of the Poisson equation. *SIAM J. Numer. Anal.* 54, 797–824.
- Gillette, A., Bajaj, C., 2011. Dual formulations of mixed finite element methods with applications. *Comput. Aided Des.* 43, 1213–1221.
- Gillette, A., Rand, A., Bajaj, C., 2012. Error estimates for generalized barycentric coordinates. *Adv. Comput. Math.* 37, 417–439.
- von Golitschek, M., Lai, M.-J., Schumaker, L.L., 2002. Error bounds for minimal energy bivariate polynomial splines. *Numer. Math.* 93, 315–331.
- Greiner, Günther, Hormann, K., 1996. Interpolating and approximating scattered 3D-data with hierarchical tensor product B-splines. In: Proceedings of Chamonix. Citeseer, p. 1.
- Hormann, K., Floater, M.S., 2006. Mean value coordinates for arbitrary planar polygons. *ACM Trans. Graph. (TOG)* 25, 1424–1441.
- Iske, A., Demaret, L., 2015. Optimally sparse image approximation by adaptive linear splines over anisotropic triangulations. In: 2015 International Conference on Sampling Theory and Applications. IEEE, pp. 463–467.
- Lai, M., 2008. Multivariate splines for data fitting and approximation. In: Neamtu, M., Schumaker, L.L. (Eds.), The Conference Proceedings of the 12th Approximation Theory. Nashboro Press, San Antonio.
- Lai, M.-J., Schumaker, L.L., 2007. Spline Functions on Triangulations, vol. 110. Cambridge University Press.
- Lai, M.-J., Slavov, G., 2016. On recursive refinement of convex polygons. *Comput. Aided Geom. Des.* 45, 83–90.
- Li, P., Adams, M.D., 2013. A tuned mesh-generation strategy for image representation based on data-dependent triangulation. *IEEE Trans. Image Process.* 22, 2004–2018.
- Li, X.-Y., Hu, S.-M., 2013. Poisson coordinates. *IEEE Trans. Vis. Comput. Graph.* 19, 344–352.
- Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D.-M., Lu, L., Yang, C., 2009. On centroidal Voronoi tessellation—energy smoothness and fast computation. *ACM Trans. Graph.* 28, 1–17.
- Lloyd, S.P., 1982. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* 28, 129–137.
- Mann, S., 1999. Cubic precision Clough-Tocher interpolation. *Comput. Aided Geom. Des.* 16, 85–88.
- Mann, S., 2006. A blossoming development of splines. *Synth. Lect. Comput. Graph. Animat.* 1, 1–108.
- Martin, S., Kaufmann, P., Botsch, M., Wicke, M., Gross, M., 2008. Polyhedral finite elements using harmonic basis functions. In: Proc. Symp. Geom. Proc., pp. 1521–1529.
- Milbradt, P., Pick, T., 2008. Polytope finite elements. *Int. J. Numer. Methods Eng.* 73, 1811–1835.
- Nivoliers, V., Lévy, B., 2013. Approximating functions on a mesh with restricted Voronoi diagrams. *Comput. Graph. Forum* 32, 83–92.
- Powell, M.J.D., 1981. Approximation Theory and Methods. Cambridge University Press.
- Powell, M.J.D., Sabin, M.A., 1977. Piecewise quadratic approximations on triangles. *ACM Trans. Math. Softw.* 3, 316–325.
- Rand, A., Gillette, A., Bajaj, C., 2014. Quadratic serendipity finite elements on polygons using generalized barycentric coordinates. *Math. Comput.* 83, 2691–2716.

- Rashid, M.M., Selimotic, M., 2006. A three-dimensional finite element method with arbitrary polyhedral elements. *Int. J. Numer. Methods Eng.* 67, 226–252.
- Sarkis, M., Diepold, K., 2009. Content adaptive mesh representation of images using binary space partitions. *Trans. Image Process.* 18, 1069–1079.
- Shewchuk, J.R., 2002. Delaunay refinement algorithms for triangular mesh generation. *Comput. Geom.* 22, 21–74.
- Sibson, R., 1980. A vector identity for the Dirichlet tessellation. *Math. Proc. Camb. Philos. Soc.* 87, 151–155.
- Sieger, D., Alliez, P., Botsch, M., 2010. Optimizing Voronoi diagrams for polygonal finite element computations. In: Proceedings of the 19th International Meshing Roundtable. Springer, pp. 335–350.
- Sukumar, N., 2013. Quadratic maximum-entropy serendipity shape functions for arbitrary planar polygons. *Comput. Methods Appl. Mech. Eng.* 263, 27–41.
- Sukumar, N., Malsch, E.A., 2006. Recent advances in the construction of polygonal finite element interpolants. *Arch. Comput. Methods Eng.* 13, 129–163.
- Sukumar, N., Tabarraei, A., 2004. Conforming polygonal finite elements. *Int. J. Numer. Methods Eng.* 61, 2045–2066.
- Tabarraei, A., Sukumar, N., 2006. Application of polygonal finite elements in linear elasticity. *Int. J. Comput. Methods* 3, 503–520.
- Talischi, C., Paulino, G.H., Pereira, A., Menezes, I.F., 2010. Polygonal finite elements for topology optimization: a unifying paradigm. *Int. J. Numer. Methods Eng.* 82, 671–698.
- Wachspress, E.L., 1975. A Rational Finite Element Basis. Academic Press, New York.
- Wicke, M., Botsch, M., Gross, M., 2007. A finite element method on convex polyhedra. *Comput. Graph. Forum* 26, 355–364.
- Zhou, T., Lai, M.-J., 2013. Scattered data interpolation by bivariate splines with higher approximation order. *J. Comput. Appl. Math.* 242, 125–140.
- Zhou, T., Han, D., Lai, M.-J., 2008. Energy minimization method for scattered data Hermite interpolation. *Appl. Numer. Math.* 58, 646–659.
- Zienkiewicz, O., Taylor, R., 2000. The Finite Element Method, 5th ed.. Butterworth-Heinemann, London.