Graphical Models 76 (2014) 522–531

Contents lists available at ScienceDirect

Graphical Models

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

Approximation by piecewise polynomials on Voronoi tessellation

Zhonggui Chen [a], Yanyang Xiao [a], Juan Cao [b,c,*]

[a] Department of Computer Science, Xiamen University, Xiamen 361005, China
[b] School of Mathematical Sciences, Xiamen University, Xiamen 361005, China
[c] Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computation, Xiamen University, Xiamen 361005, China

ARTICLE INFO

Article history:
Received 2 March 2014
Received in revised form 30 March 2014
Accepted 2 April 2014
Available online 18 April 2014

Keywords:
Polynomial approximation
Voronoi tessellation
Optimization
Image approximation

ABSTRACT

We propose a novel method to approximate a function on 2D domain by piecewise polynomials. The Voronoi tessellation is used as a partition of the domain, on which the best fitting polynomials in $\mathcal{L}^2$ metric are constructed. Our method optimizes the domain partition and the fitting polynomials simultaneously by minimizing an objective function indicating the approximation quality. We also provide the explicit formula of the gradient of the objective function, which makes an efficient gradient-based algorithm workable for the function minimization. We conduct several experiments to demonstrate the efficacy of our new approach for generating piecewise polynomial approximations of analytic functions and color images.

1. Introduction

The problem of approximating a given function concisely is not only of importance in applied mathematics but also at the core of many applications in computer graphics. Generally, a function approximation problem seeks a function in a small candidate space to closely match an arbitrarily given target function. The space of polynomials is of particular interest in function approximation due to their simplicity and flexibility [1]. In this paper, we are interested in seeking the optimal piecewise polynomial approximation of a scalar function defined in 2D domain and exploring its application to image approximation.

For piecewise polynomial approximation, the quality of the approximation resultant is affected by the domain partition and degree of the approximation space. The error of the resultant indicates how well it approximates the target function, hereinafter, we use the most commonly used $\mathcal{L}^2$ norm as the error measure. Once the number of partitions and the degree of approximation space are specified, the approximation problem becomes how to divide the domain into contiguous patches. It is because that as long as the partition of the domain is given, the optimal polynomial approximation can be simply obtained by solving a linear least-squares problem. Thus, generating a good partition of the domain is the key challenge to solving the piecewise approximation problem.

It has be shown in [2] that it is NP-hard to decide whether the surfaces that correspond to graphs of bivariate functions can be approximated by polyhedral surfaces within a prespecified error. A common heuristic is to recursively subdivide the patch with maximum approximation error into smaller patches, which usually results in an excessive number of patches [3]. If a discrete representation of the domain is given, such as the images in [4] and the triangular meshes in [5,6], a flooding method can be used to cluster pixels or triangles into partitions based on minimizing the approximation error of given polynomials.

Our method works on continuous domain directly. In order to simplify the computation of finding an appropriate

* Corresponding author at: School of Mathematical Sciences, Xiamen University, B511, Xiamen, Fujian 361000, China.
E-mail addresses: chenzhonggui@xmu.edu.cn (Z. Chen), jxndxyy@gmail.com (Y. Xiao), juancao@xmu.edu.cn (J. Cao).

http://dx.doi.org/10.1016/j.gmod.2014.04.006
1524-0703/

domain partition for the purpose of optimal approximation, we restrict ourselves to the Voronoi tessellation, determined solely upon the sites (or generators) on the domain. The approximation error indirectly relies on the positions of these sites, which can be minimized by an efficient gradient-based optimization method. This paper is a generalization of the piecewise constant polynomial approximation method proposed by Nivoliers and Lévy in [7]. The specific contributions of the paper include.

- We propose a variational scheme for function approximation with piecewise polynomials of an arbitrary degree defined over Voronoi tessellation.
- The formula of the gradient of the new energy function can be explicitly derived, which makes it possible to employ an efficient optimization method.
- Benefiting from using higher degree approximation, our method is capable of generating visually pleasant results in image approximation.

After a short review of the related work in Section 1.1, we describe our method in Section 2. Experimental results are shown in Section 3. And we draw our conclusion in Section 4.

1.1. Related work

1.1.1. Polynomial approximation

The problem of approximating a function by polynomials has been investigated extensively, and a number of efficient algorithms have been proposed [1]. Most previous works focus on the approximation of univariate function. The more general problem of approximating functions by multivariate polynomials is complicated and is still an active subject in the area of research [8]. Generally, if an orthogonal basis is available, the least-squares approximation problem can be easily coped with [9]. The construction of univariate orthogonal polynomials can be done by the Gram-Schmidt process. Generalizing from univariate case to multivariate case poses much harder mathematical and algorithmic challenges. The bivariate orthogonal polynomials construction has been achieved only in some special cases, such as on regular hexagon domain [10] and triangular domain [11]. The construction of multivariate orthogonal polynomials on any given domain with an arbitrary shape is wide open. Thus, we use the power polynomial basis in this paper. Our algorithm achieves good performance and numerical stability in our experiments, when low degree polynomial is used.

1.1.2. Variational surface approximation

Cohen-Steiner et al. [5] propose a versatile variational framework for constructing piecewise linear approximations of mesh surfaces. The approximation error is minimized by alternating optimizing the partitions of the domain and finding an optimal fitting plane for each segment. A distortion minimization flooding method is proposed to cluster mesh facets into partitions. Several variants of Cohen-Steiner et al.'s method have been proposed to include more types of geometric primitives for better approximation. Wu and Kobbelt [12] introduce spheres, cylinders and rolling-ball blend patches as basic primitives. In [13], the partition of a mesh is optimized by fitting to ellipsoidal surface regions. Yan et al. [6] extend [5]'s method for general quadric surfaces. Our method shares some similarities with [5]'s framework. One important difference is that, our method optimizes the segmentation and the fitting primitives simultaneously, by imposing the condition that the partition of the domain is a Voronoi tessellation. Optimization of the partition of the domain is naturally included in our optimization process.

1.1.3. Image approximation

There has been a lot of research on the automatic extraction of effective geometric representations from images. The vector curves and region primitives are usually used to fit the clusters of pixels obtained by segmentation or edge detection algorithms. A popular geometric representation of image is the linear approximation [14–16], which is defined by a triangulated subset of the source image pixel. However, most previous methods do not explicitly take image discontinuities into consideration. Due to the C^0 continuity of the linear approximation, more vertices of the triangulation have to be placed along the both sides of edges in the image to capture the sharp change of colors. Lecot and Lévy [4] present a method for vectorizing raster images by using first- or second-order gradients. It generalizes Cohen-Steiner et al.'s variational framework [5] to image-processing setting. A flooding method is developed to cluster pixels into large regions, to which the high-order functions are fit. The regions could be with arbitrary shapes and zigzag boundaries. In our method, we decompose the domain by Voronoi tessellation, which is simple in shape and independent of the image resolution.

2. Piecewise polynomial approximation

2.1. Objective function

Assume that $\{P_1(\mathbf{x}),\ldots,P_m(\mathbf{x})\}$ is a basis of a polynomial space $\mathcal{P}_n$ of degree equal to a given integer n. Then any polynomial $Q(\mathbf{x})$ in $\mathcal{P}_n$ can be represented as a linear combination of the basis $Q(\mathbf{x})=\sum_{j=1}^{m}c_jP_j(\mathbf{x})$. Let $f(\mathbf{x})$ be a function defined over a compact 2D domain Ω. Suppose that $\mathcal{T}=\{\Omega_i\subset\Omega, i=1,\ldots N \mid \Omega_i\bigcap\Omega_j=\emptyset,\forall i\neq j;\bigcup_{i=1}^{N}\Omega_i=\Omega\}$ is a partition of the domain Ω. Then over each subdomain Ω_i, function $f(\mathbf{x})$ can be approximated by a polynomial Q_i in $\mathcal{P}_n$. Our goal is to find the optimal piecewise polynomial approximation of $f(\mathbf{x})$ in the $\mathcal{L}^2$ norm, i.e., the best approximation which minimizes the error as follows:

$$E(\{Q_i\in\mathcal{P}_n\}_{i=1}^{N},\mathcal{T})=\sum_{i=1}^{N}\int_{\Omega_i}|f(\mathbf{x})-Q_i(\mathbf{x})|^2\mathrm{d}\mathbf{x}. \quad (1)$$

The optimal solution of the above energy function includes two components: the optimal partition of the domain and the optimal polynomial approximation on each region Ω_i. In this paper, we are going to restrict ourselves to the case where $\mathcal{T}$ is a Voronoi tessellation of a site set $X=\{\mathbf{x}_i\}_{i=1}^{N}$, i.e., Ω_i is a Voronoi cell of site $\mathbf{x}_i\in X$.

Notice that, for a fixed partition $\mathcal{T}$, the optimal polynomial approximation $Q_i^*(\mathbf{x})$ is a minimizer of the following least-squares problem:

$$Q_i^*(\mathbf{x}) = \arg\min_{Q(\mathbf{x})\in\mathcal{P}_n} \int_{\Omega_i} |f(\mathbf{x}) - Q(\mathbf{x})|^2 d\mathbf{x}.$$

The coefficients $(c_1^*, \ldots, c_m^*)$ in the polynomial $Q_i^*(\mathbf{x})$ can be obtained by solving a linear system:

$$\sum_{j=0}^{m} c_j^* \int_{\Omega_i} P_j(\mathbf{x})P_k(\mathbf{x})d\mathbf{x} = \int_{\Omega_i} P_k(\mathbf{x})f(\mathbf{x})d\mathbf{x}, k = 1, \ldots, m. \quad (2)$$

Thus, $Q_i^*(\mathbf{x})$ relies only on the tessellation $\mathcal{T}$. On the other hand, the Voronoi tessellation $\mathcal{T}$ is only determined by the site positions X. The objective function in Eq. (1) therefore can be reconsidered as a function of X:

$$E(X) = \sum_{i=1}^{N} \int_{\Omega_i} |f(\mathbf{x}) - Q_i^*(\mathbf{x})|^2 d\mathbf{x}. \quad (3)$$

In the rest of this section, we will try to find the optimal positions of the sites which minimize the objective function.

2.2. Gradient of $E(X)$

Let J_i denote the indices of sites whose Voronoi cells are adjacent to Ω_i. To deduce the derivative of $E(X)$ with respect to $\mathbf{x}_i$, we only need to consider the items involving $\mathbf{x}_i$ in $E(X)$. Hence, the derivative becomes

$$\frac{\partial E(X)}{\partial \mathbf{x}_i} = \frac{\partial}{\partial \mathbf{x}_i} \sum_{j\in J_i \bigcup \{i\}} \int_{\Omega_j} |f(\mathbf{x}) - Q_j^*(\mathbf{x})|^2 d\mathbf{x}. \quad (4)$$

Notice that, $\mathbf{x}_i$ appears in both the integrand and the integral domain of the integral terms. Therefore, we should also take the variation of the integral domain in differentiation into consideration. Now let us recall the general Leibniz rule [17] to simplify this formula. Suppose that D_t is a 2D domain whose boundary ∂D_t changes smoothly with respect to t, and $g(\mathbf{x}, t), \mathbf{x} \in D_t$ is a function defined over D_t. Let the velocity vector of a point on the domain boundary ∂D_t be denoted by $\mathbf{v} = \partial\mathbf{x}/\partial t$ and $\mathbf{n}$ be the outward unit normal vector at the boundary. Then the general Leibniz rule [17] states:

$$\frac{d}{dt}\int_{D_t} g(\mathbf{x}, t)d\mathbf{x} = \int_{D_t} \frac{\partial g(\mathbf{x}, t)}{\partial t} d\mathbf{x} + \int_{\partial D_t} g(\mathbf{x}, t)\mathbf{v}\cdot\mathbf{n} ds,$$

where ds is the element of arc length on the closed boundary curve ∂D_t.

Applying the general Leibniz rule to Eq. (4), we have

$$\frac{\partial E(X)}{\partial \mathbf{x}_i} = \sum_{j\in J_i \bigcup \{i\}} \int_{\Omega_j} \frac{\partial}{\partial \mathbf{x}_i} |f(\mathbf{x}) - Q_j^*(\mathbf{x})|^2 d\mathbf{x} + \sum_{j\in J_i} \int_{\Omega_{ij}} \left(|f(\mathbf{x}) - Q_i^*(\mathbf{x})|^2 - |f(\mathbf{x}) - Q_j^*(\mathbf{x})|^2\right) \frac{\partial \mathbf{x}}{\partial \mathbf{x}_i}\mathbf{n} ds,$$

where $\Omega_{ij} = \partial\Omega_i \bigcap \partial\Omega_j$ is the boundary shared by Ω_i and Ω_j. Let us first consider the integration in the first term in the above equation. From Eq. (2), we have that $Q_i^*(\mathbf{x})$ satisfies

$$\int_{\Omega_i} (f(\mathbf{x}) - Q_i^*(\mathbf{x}))P_k(\mathbf{x})d\mathbf{x} = 0, \quad k = 1, \ldots, m.$$

Thus,

$$\int_{\Omega_j} \frac{\partial}{\partial \mathbf{x}_i} |f(\mathbf{x}) - Q_j^*(\mathbf{x})|^2 d\mathbf{x} = 2\sum_{k=1}^{m} \int_{\Omega_j} (f(\mathbf{x}) - Q_j^*(\mathbf{x}))P_k(\mathbf{x}) \frac{\partial c_k^*}{\partial \mathbf{x}_i} d\mathbf{x} = 0.$$

Now let us turn to the evaluation of the term $\partial\mathbf{x}/\partial\mathbf{x}_i\mathbf{n}$ on the segment Ω_{ij}. We notice that the point $\mathbf{x}$ on Ω_{ij} satisfies

$$\left(\mathbf{x} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2}\right)\cdot(\mathbf{x}_j - \mathbf{x}_i) = 0.$$

By differentiating the above equation with respect to $\mathbf{x}_i$, we get

$$\frac{\partial \mathbf{x}}{\partial \mathbf{x}_i}(\mathbf{x}_j - \mathbf{x}_i) = \mathbf{x} - \mathbf{x}_i.$$

Thus, we have $\partial\mathbf{x}/\partial\mathbf{x}_i\mathbf{n} = (\mathbf{x} - \mathbf{x}_i)/|\mathbf{x}_j - \mathbf{x}_i|$. And the formula for the derivative of $E(X)$ with respect to $\mathbf{x}_i$ is simplified to

$$\frac{\partial E(X)}{\partial \mathbf{x}_i} = \sum_{j\in J_i} \int_{\Omega_{ij}} \left(|f(\mathbf{x}) - Q_i^*(\mathbf{x})|^2 - |f(\mathbf{x}) - Q_j^*(\mathbf{x})|^2\right) \frac{\mathbf{x} - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} ds. \quad (5)$$

Note that our work naturally generalizes the work of [7], as our objective function differs from the objective function in [7] only in the approximation function space. In particular, our approximation function space is piecewise polynomials with an arbitrary order. On contrary, the approximation function space is restricted to piecewise constant functions. Consequently, the formula of gradient (5) for the higher order approximation space is also analogues to the gradient formula in [7]. As both [7] and our method use a gradient-based solution mechanism, we will show in next section that the optimization frameworks are basically the same, with a slightly different approximation computation in each iteration step.

2.3. Solution mechanism

Efficient optimization solvers such as quasi-Newton methods [18] require C^2 continuity of the objective function. Whereas, the objective function $E(X)$ in Eq. (3) can only be C^0 when the function $f(\mathbf{x})$ is discontinuous. Furthermore, it is shown that, the magnitudes of the derivatives of $E(X)$ with respect to different variables may vary greatly even in the piecewise constant approximation [7]. Thus, the conventional gradient descent approach is not suitable for the case in our context either.

An improved gradient-based method has been designed for effectively optimizing the objective function $E(X)$ in the piecewise constant approximation case in [7]. Here, we follow this method with making a slight modification on the step lengths as follows: Starting from an initialization, all the positions of the sites are optimized iteratively as:

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

where j is the index of the current iteration, and $\delta_i^{(j)}$ is the step length for $\mathbf{x}_i$. Notice that, the descent direction here is obtained by normalizing each component of the gradient respectively. The descent direction is therefore no longer the gradient direction, and the optimization method here is no longer a gradient decent method. The step length $\delta_i^{(j)}$, gradually decreasing to zero, are controlled by the following formula

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

where $J_{\max}$ is the specified maximum number of iterations and $\delta_i^{(0)}$ is the initial step length for site $\mathbf{x}_i$. In the original method, $\delta_i^{(0)}$ for all sites are set to the same value (2% of the scene bounding box diagonal). The uniform initial step length limits the method to the problem with a relatively fixed site number and even distribution. To overcome this, the initial step length is set to $k\sqrt{\text{area}(\Omega_i)}$, where Ω_i is the Voronoi cell of the site i and k is a scaling parameter.

An appropriate magnitude of the scaling parameter is important. If it is too small, the sites will stay around at their initial positions, while a too large step length would amount to redistributing the sites over the domain. With extensive experiments, we found that $k = 0.5$ is a good choice for producing satisfying results. Experiments also show that the above method is more efficient than several gradient-based methods, including conventional gradient descent method, conjugate gradient method, and L-BFGS method [18], in optimizing the objective function $E(\mathbf{X})$. The pseudo-code in Algorithm 1 illustrates our algorithm.

Algorithm 1. Piecewise polynomial approximation

Input: a function $f(\mathbf{x})$ on Ω, an initial site set $\mathbf{X}$ on Ω, a polynomial basis $\{P_1(\mathbf{x}), \ldots, P_m(\mathbf{x})\}$ and the max iteration number $J_{\max}$

Output: a Voronoi tessellation $\mathcal{T}$ corresponding to a new site distribution $\mathbf{X}$, and a set of polynomials $\{Q_1(\mathbf{x}), \ldots, Q_N(\mathbf{x})\}$ on $\mathcal{T}$

$\mathbf{X}^{(0)} \leftarrow \mathbf{X}, \quad j \leftarrow 0$
Compute Voronoi tessellation $\mathcal{T} = \{\Omega\}_{i=1}^N$ of $X^{(0)}$
$\delta_i^{(0)} \leftarrow \sqrt{area(\Omega_i)}/2, \quad i = 1, \ldots, N$
while $j < J_{\max}$ **do**
  Compute Voronoi tessellation $\mathcal{T}$ of $\mathbf{X}^{(j)}$
  Compute the optimal polynomial approximations $\{Q_1(\mathbf{x}), \ldots, Q_N(\mathbf{x})\}$ over $\mathcal{T}$
  Compute the gradient $\nabla E(X)$
  for each $i \in \{1, \ldots, N\}$ **do**
    $\delta_i^{(j)} = \delta_i^{(0)} (\frac{1}{2})^{\frac{j}{J_{\max}-j}}$
    $\mathbf{x}_i^{(j+1)} \leftarrow \mathbf{x}_i^{(j)} - \delta_i^{(j)} \frac{\partial E(X)}{\partial \mathbf{x}_i} \Big/ \left| \frac{\partial E(X)}{\partial \mathbf{x}_i} \right|$
  end for
  $j \leftarrow j + 1$
end while

Now let us turn to the initialization of the algorithm. A good initialization let our local search method converge fast. In our algorithm, the initialization of sites $\mathbf{x}_i^{(0)}$ are "good" if they are situated sufficiently close to a minimizer of the objective function in Eq. (3). Notice that, a global minimizer of the objective function in Eq. (3) tends to have a uniform cumulated approximation errors in each Voronoi region. Inspired by this, we propose a greedy method for site initialization as follows: Site is sequentially added until a user-specified point budget is reached, and the newly added site is randomly sampled from the Voronoi region with maximum approximation error.

3. Results and applications

The algorithm was implemented and tested using C++. The CGAL library [19] is used to compute the Voronoi tessellation (with a conversion from its dual Delaunay diagram). Our algorithm has been tested extensively on function approximations by using basis of polynomial space of different degrees. Specifically, the polynomial bases $\{1, x, y, xy, x^2, y^2\}, \{1, x, y\}$ and $\{1\}$ have been used for piecewise quadratic, linear and constant approximations.

The integrals involved in the algorithm are numerically computed by using quadrature rules. We used different quadrature rules for the computation of the integrals, based on the considerations of efficiency and accuracy. In particular, if the integrand is smooth, each Voronoi region is first decomposed to triangles by connecting the centroid with its vertices, then a three-point Gaussian quadrature rule (three-point *Dunavant* rule) for the triangle [20] is used for the domain integration. When the integrand is discontinuous, as in image approximation, each triangle is repeatedly subdivided into smaller sub-triangles by 1–4 split method until the area of sub-triangle is equal or smaller than five pixels, then the one-point *Dunavant* rule is used. For the boundary integration, the classical two-point Gaussian quadrature rule and one-point Gaussian quadrature rule are used for the smooth and non-smooth integrands, respectively.

3.1. Testing functions

The first test case is the function

$$f(x, y) = 1 - \sqrt{x^2 + y^2}, \quad -\frac{\sqrt{2}}{2} \leqslant x, \quad y \leqslant \frac{\sqrt{2}}{2},$$

and piecewise constant and piecewise linear approximations with 500 sites. The optimized Voronoi tessellations for both approximations are shown in Fig. 2. Notice that, the orientations of the cells align with the isoline and the gradient direction of the function in the piecewise constant and linear approximations, respectively.

The second test case is a paraboloid

$$f(x, y) = x^2 + y^2, \quad -1 \leqslant x, \quad y \leqslant 1,$$

and a piecewise linear approximation. The result Voronoi tessellation and the approximating surface are shown in

(a) Original Image (b) Piecewise constant approximation

(c) Piecewise linear approximation (d) Piecewise quadratic approximation

Fig. 1. Image approximation on a Voronoi tessellation with 1000 patches by piecewise polynomials.

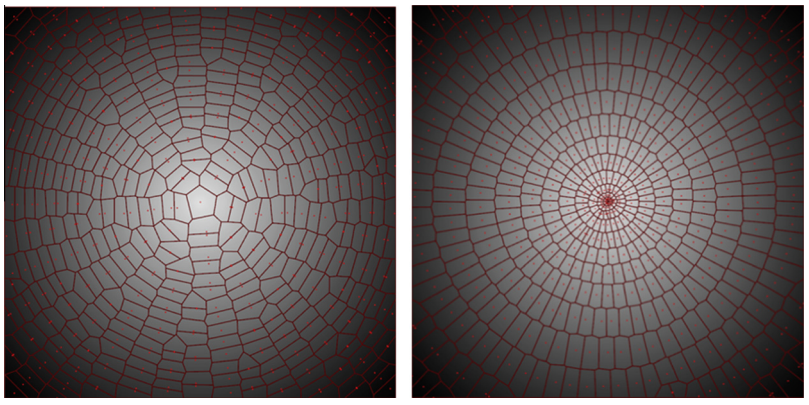

Fig. 2. Approximations of the function $1-\sqrt{x^2+y^2}$ on square. The domain is color-coded by the function value. Left: the result Voronoi tessellation in piecewise constant approximation. Right: the result Voronoi tessellation in piecewise linear approximation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 3. It is worth pointing out that, in this special case of linear approximation, our method generate a nearly hexagonal tiling, which is similar to the result yielded by centroidal Voronoi tessellation (CVT) method [21]. Our objective function and the CVT function differ in both the distance measures and the linear functions. In particular, the objective function of CVT method [21] is defined as the distance between the function and its linear approximant in

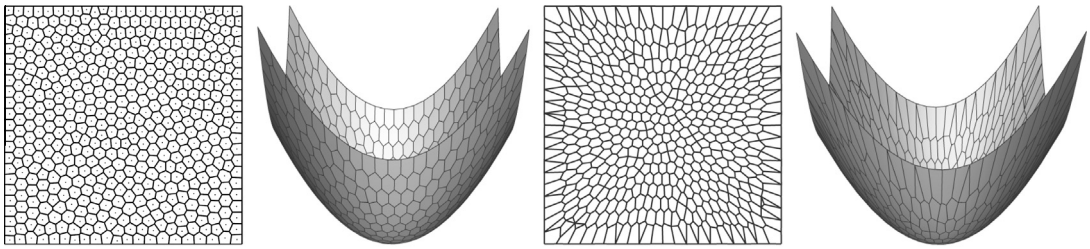

Fig. 3. Approximation of the quadratic function $f(x,y) = x^2 + y^2$ by piecewise linear polynomials on square. (a) and (b) The result Voronoi tessellation and the fitting surface by our method; (c) and (d) the result domain tessellation and the fitting surface by VSA method [5].

the $\mathcal{L}^1$ norm, which actually measures the volume between the paraboloid and a circumscribing piecewise linear function (i.e., the local tangent plane) [22]. Whereas, we use the $\mathcal{L}^2$ norm as the distance measure in our objective function definition, and the linear function is constrained to the optimal linear approximation.

Despite the use of different distance measure and approximating polynomials, both methods yield similar resultant in the sense that the tessellation is nearly a hexagonal tiling, as can be observed in Fig. 3. This phenomena can be explained by the fact that, with respect to $\mathcal{L}^p$ metric, any optimal linear approximation of a function have the same element's aspect ratio $|\frac{\sigma_{min}}{\sigma_{max}}|$, where σ_{min} and σ_{max} are the minimum and maximum of eigenvalues of its Hessian [23], respectively. Our method can be viewed as a more general method than CVT method in terms of generating tessellations with a variety of cell shapes. In Fig. 3, we also compare our method with the variational shape approximation (VSA) method by [5]. It can be obviously observed in Fig. 3 that our method generates better result than VSA method in the sense that the tessellation generated by our method is more uniform than the one by VSA method.

The third test case is a discontinuous function

$$f(x,y) = \begin{cases} \sin(\pi(x+1/2))\cos(\pi y), & -\frac{1}{2} \leqslant x, \quad y \leqslant \frac{1}{2}, \\ e^{xy} - 1, & \text{others} \end{cases}$$
$$-1 \leqslant x, \quad y \leqslant 1,$$

and a piecewise linear approximation. For the approximation methods based on triangulations [15,24], additional steep faces are needed to connect the two disconnected pieces of the function. Unlike those previous methods, our method is capable of representing the discontinuity of the function faithfully, since the resultant Voronoi regions automatically align along the line where the function is discontinuous, see in Fig. 4.

3.2. Image approximation

Our method can be directly applied to the approximation of gray-scale image, which can be taken as a discontinuous function. For a RGB color image, its red, green and blue channels can be separated from the image and considered as three independent functions $r(\mathbf{x}), g(\mathbf{x})$, and $b(\mathbf{x})$, respectively. In order to simultaneously approximate these three functions in the same image domain tessellation, we modify the objective function in Eq. (3) to

$$E(X) = \sum_{i=1}^{N} \int_{\Omega_i} \Big(|\, r(\mathbf{x}) - R_i^*(\mathbf{x})|^2 + |\, g(\mathbf{x}) - G_i^*(\mathbf{x})|^2 + |\, b(\mathbf{x}) - B_i^*(\mathbf{x})|^2 \Big) \mathrm{d}\mathbf{x}, \tag{6}$$

where $R_i^*(\mathbf{x}), G^*(\mathbf{x})$, and $B^*(\mathbf{x})$ are the optimal polynomial approximations of $r(\mathbf{x}), g(\mathbf{x})$, and $b(\mathbf{x})$ on Ω_i, respectively. Algorithm 1 can still be applied to the optimization of the modified objective function as its derivatives can be evaluated in the same fashion in Section 3.2.

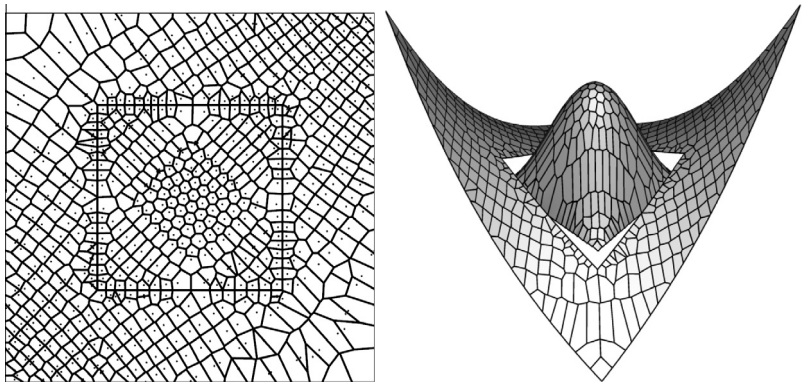

Fig. 4. Approximation of a discontinuous function. Left: the result Voronoi tessellation. Right: the fitting surface by piecewise linear approximation.

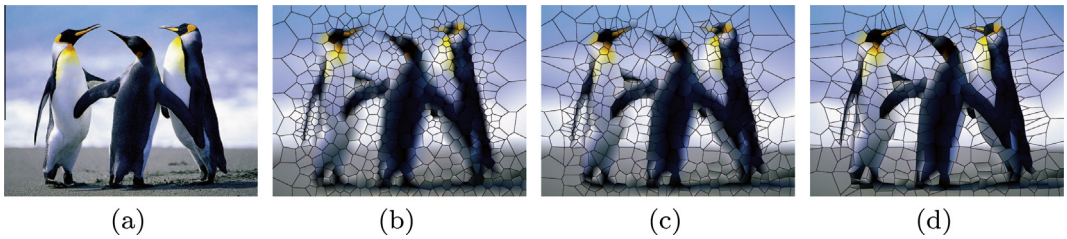

Fig. 5. Algorithm overview of color image approximation using linear polynomials. (a) Original image; (b) initialization; (c) after 10 iterations; (d) final result after 150 iterations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The intermediate results generated in our algorithm in image approximation are shown in Fig. 5. The initial sites are sequentially added into the Voronoi region with maximum approximation error, as shown in Fig. 5(b). After a few iterations, the positions of the sites are progressively adjusted to reduce the approximation error. And the boundaries of Voronoi regions align with the feature lines of the image in most case.

3.2.1. Post-processing

As our method only focuses on minimizing the approximation error over the whole image domain in the least-squares sense, there may exists a small number of Voronoi regions whose approximation errors are still severe after optimization. These Voronoi regions may cross the feature lines, cover fine details or occupy large areas. However, only Voronoi regions crossing feature lines lead to obvious artifacts in the result image. As shown in Fig. 5(b), the Voronoi regions partially overlap with two different objects of the original image in the same time, hence the resultant image is locally visually unpleasant. We here introduce a post-processing step to locate and destroy Voronoi regions crossing feature lines to eliminate the artifacts.

As a Voronoi region with a large approximation error does not necessarily leads to obvious artifacts, we need to introduce another metric, according to which, we can distinguish the Voronoi regions crossing feature lines. For the sake of simplicity, we convert a color image to gray-scalar one and define the color-bias of a Voronoi region as follows: the pixels within this Voronoi region are divided into two groups according to whether its gray value is bigger or smaller than the average color, then the color-bias of the Voronoi cell is defined as a

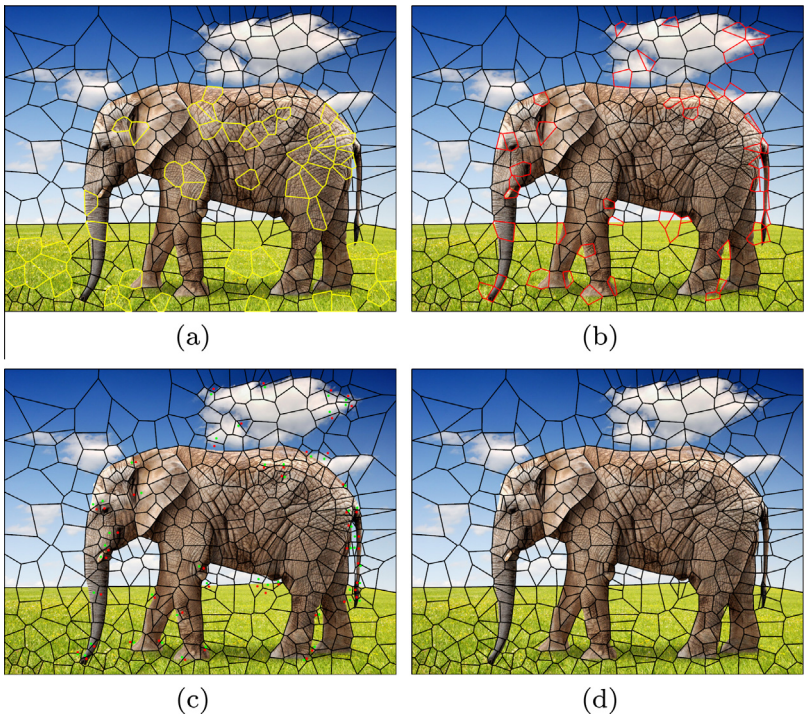

Fig. 6. Detecting the Voronoi regions crossing feature lines. (a) detected regions according to approximation error, marked in yellow color; (b) detected regions according to the color-bias, marked in red color; (c) intermediate result with newly inserted sites; and (d) final result with newly inserted sites being optimized. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 7. Image approximation by piecewise polynomials with 1000 patches.

Table 1
The statistics of the running times.

Fig.	Image res.	#Sites	Order	J_{max}	Time (s)
1	600 × 600	1000	Constant	150	120.787
			Linear		143.296
			Quadratic		145.178
2	N.A.	500	Constant	1000	23.593
			Linear		24.904
3	N.A.	500	Linear	300	7.392
4	N.A.	680	Linear	300	10.51
5	1024 × 768	500	Linear	150	274.911
7	512 × 512	1000	Constant	150	104.861
			Linear		113.814
			Quadratic		114.815
8	300 × 208	500	Quadratic	150	33.97
		1000			59.263
		2000			171.224

combination of the absolute difference of the average colors of the two pixel groups and the distance between the centroids of the two pixel groups. Roughly, a region with a big color-bias is likely to lean toward two colors, that is, this region crosses a feature line of the image. Having the color-bias in hand, the regions with artifacts can easily be identified according to the value of the color-bias. As shown in Fig. 6, the Voronoi regions with the 50 largest color-biases are marked in red. As a comparison, the Voronoi regions with the 50 largest approximation errors are marked in yellow.

Once a Voronoi region with artifacts is detected, we proceed to divide it into two parts by removing the original site and inserting two points at the centroids of the two pixel groups, respectively. Then we run Algorithm 1 again to optimize the positions of the newly added sites, as

Fig. 8. Multi-resolution approximation by piecewise quadratic polynomials with 500 (left), 1000 (middle) and 2000 (right) patches.

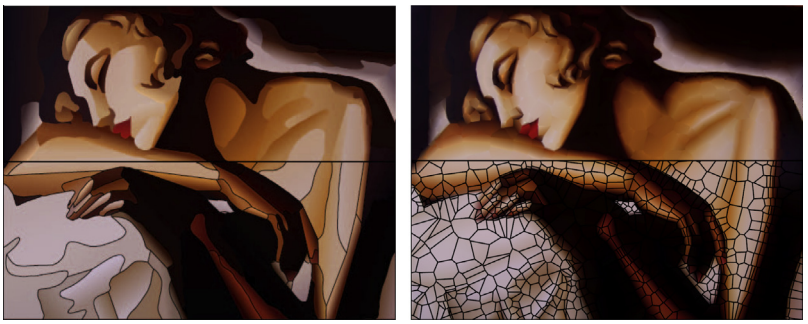

Fig. 9. Comparison with the image vectorization method by [4]. Left: result from [4] with 180 primitives. Right: linear approximation result by our method with 1500 patches.

shown in Fig. 6. Experiments show that, our method can capture most feature lines of the image after the "insert-optimize" process is completed.

Some approximation results using constant, linear, and quadratic polynomial bases are shown in Figs. 1, 7 and 8. The piecewise constant approximation proposed in [7] creates mosaic-like effect, where color abruptly changes between neighboring patches. By using high-order polynomial basis, the approximation quality can significantly be improved, as illustrated in Figs. 1 and 7. A multi-resolution approximation by piecewise quadratic polynomials is shown in Fig. 8.

All the experiments are conducted on a PC with Intel I5 3.1 GHz CPU and 4.0 GB RAM. Our optimization process usually takes about a few seconds to several minutes, depending on the site and iteration numbers and the image resolution. The statistics of the running times of the examples are given in Table 1.

4. Conclusion

In this paper, we have articulated a novel method that computes a piecewise polynomial approximation of a function on 2D domain. We extend Nivoliers and Lévy's method [7] to higher order approximations. The domain partition is constrained to be a Voronoi tessellation of a set of sites. The objective function which measures the quality of the approximation only depends on the positions of the sites. We derive the close-form expression of the gradient of the objective function. Thus an efficient optimization method can be adopted for the minimization. Experiments are conducted to show its efficacy in approximating analytic functions, and its applicability to color image approximation.

4.1. Limitations and Future work

Our formulation of the approximation problem applies to polynomials of any order. However, as we use power polynomial basis in this paper, the coefficient matrix of the linear system in Eq. (2) will become ill-conditioned when high order basis are employed, which leads to numerical instabilities in the computation. We therefore plan to further refine our algorithm by integrating other stable bases, such as the Bernstein polynomials and other bi-variable splines [25], which although do not possess orthogonality property but have shown numerical stability in practice.

Another limitation is that our method only focus on piecewise smooth approximations. The algorithm is very practical to applications such as image approximation. However, when global continuity in the approximation result is desired, our method would be inapplicable. Due to the constraint of Voronoi tessellation, our method usually needs much more samples to achieve the similar visual fidelity to the original image compared to other image vectorization methods, such as the Ardeco method [4] as shown in Fig. 9. To achieve satisfactory vectorized images, our method need be further refined by paying more attentions to region representation and feature preservation.

In this paper, we only consider the partition of the planer domain with the Euclidean metric. For future work, we will extend our method to surface domain, where the geodesic metric [26] can be used to build the Voronoi tessellation. It is also promising to incorporate the geodesic distance on the image domain [27,28], which will benefit the generation of structure-sensitive tessellations of the image domain. As it has been pointed out in the second testing function in Section 3.1 that our method generates result similar to the CVT method in the linear approximation case. Given the widespread use of CVT method, it is worth to further study the relation between the objective functions in our method and in CVT method. And we would also explore the potential applications of our method, such as mesh generation, superpixel segmentation, image vectorization and so on.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. The work is supported by NSFC (61100107), NSFC (61100105) and Natural Science Foundation of Fujian Province of China (2012J01291 and 2011J05007).

References

[1] M.J.D. Powell, Approximation Theory and Methods, Cambridge University Press, 1981.
[2] P.K. Agarwal, S. Suri, Surface approximation and geometric partitions, in: Proceedings of the Fifth Annual ACM-SIAM

Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 1994, pp. 24–33.
[3] J. Prentice, Range and domain partitioning in piecewise polynomial approximation, Stud. Math. Sci. 2 (2) (2011) 67–77.
[4] G. Lecot, B. Lévy, Ardeco, Automatic region detection and conversion, in: Proceedings of the 17th Eurographics conference on Rendering Techniques, Eurographics Association, 2006, pp. 349–360.
[5] D. Cohen-Steiner, P. Alliez, M. Desbrun, Variational shape approximation, ACM Transactions on Graphics (TOG), vol. 23, ACM, 2004, pp. 905–914.
[6] D.-M. Yan, W. Wang, Y. Liu, Z. Yang, Variational mesh segmentation via quadric surface fitting, Comput.-Aided Des. 44 (11) (2012) 1072–1082.
[7] V. Nivoliers, B. Lévy, Approximating functions on a mesh with restricted Voronoï diagrams, Comput. Graphics Forum 32 (5) (2013) 83–92.
[8] M. Amirfakhrian, Best approximation of multivariate functions in L1 and L2 by optimization, Math. Sci. 4 (2) (2010) 205–219.
[9] G. Fedele, A. Ferrise, Explicit solution of the finite time L2-norm polynomial approximation problem, Appl. Math. Comput. 217 (21) (2011) 8354–8359.
[10] C.F. Dunkl, Orthogonal polynomials on the hexagon, SIAM J. Appl. Math. 47 (2) (1987) 343–351.
[11] R.T. Farouki, T.N. Goodman, T. Sauer, Construction of orthogonal bases for polynomials in bernstein form on triangular and simplex domains, Comput. Aided Geometric Des. 20 (4) (2003) 209–230.
[12] J. Wu, L. Kobbelt, Structure recovery via hybrid variational surface approximation, Comput. Graphics Forum 24 (2005) 277–284.
[13] P.D. Simari, K. Singh, Extraction and remeshing of ellipsoidal representations from mesh data, in: Graphics Interface, 2005, pp. 161–168.
[14] N. Dyn, D. Levin, S. Rippa, Data dependent triangulations for piecewise linear interpolation, IMA J. Numer. Anal. 10 (1) (1990) 137–154.
[15] O. Kreylos, B. Hamann, On simulated annealing and the construction of linear spline approximations for scattered data, IEEE Trans. Visual. Comput. Graphics 7 (1) (2001) 17–31.
[16] D. Su, P. Willis, Image interpolation by pixel-level data-dependent triangulation, Computer Graphics Forum, vol. 23, Wiley Online Library, 2004, pp. 189–201.
[17] H. Flanders, Differentiation under the integral sign, Am. Math. Monthly 80 (6) (1973) 615–627.
[18] J. Nocedal, S.J. Wright, Numerical Optimization, 2nd Edition., Springer, 2006.
[19] A. Fabri, CGAL-the computational geometry algorithm library, in: Proceedings of 10th International Meshing Roundtable, 2001, pp. 137–142.
[20] D.A. Dunavant, High degree efficient symmetrical Gaussian quadrature rules for the triangle, Int. J. Numer. Methods Eng. 21 (1985) 1129–1148.
[21] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, SIAM Rev. 41 (1999) 637–676.
[22] P. Alliez, D. Cohen-Steiner, M. Yvinec, M. Desbrun, Variational tetrahedral meshing, ACM Trans. Graphics (Proc. SIGGRAPH) 24 (3) (2005) 617–625.
[23] J.R. Shewchuk, What is a good linear element? Interpolation, conditioning, and quality measures, in: Proc. the 11th International Meshing Roundtable, 2002, pp. 115–126.
[24] L. Chen, Mesh smoothing schemes based on optimal Delaunay triangulations, in: Proc. 13th International Meshing Roundtable, 2004, pp. 109–120.
[25] J. Cao, X. Li, Z. Chen, H. Qin, Spherical DCB-spline surfaces with hierarchical and adaptive knot insertion, IEEE Trans. Visual. Comput. Graphics 18 (8) (2012) 1290–1303.
[26] Y. Liu, Z. Chen, K. Tang, Construction of iso-contours, bisectors, and voronoi diagrams on triangulated surfaces, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1502–1517.
[27] X. Bai, G. Sapiro, A geodesic framework for fast interactive image and video segmentation and matting, in: IEEE 11th International Conference on Computer Vision, 2007, ICCV 2007, IEEE, 2007, pp. 1–8.
[28] P. Wang, G. Zeng, R. Gan, J. Wang, H. Zha, Structure-sensitive superpixels via geodesic distance, Int. J. Comput. Vision 103 (1) (2013) 1–21.