Reconstructing High-Order Surfaces for Meshing

Xiangmin Jiao* and Duo Wang

Department of Applied Mathematics, Stony Brook University, Stony Brook, NY 11794

Summary. We consider the problem of reconstructing a high-order surface from a given surface mesh. This problem is important for many meshing operations, such as generating high-order finite elements, mesh refinement, mesh smoothing and mesh adaptation. We introduce two methods, called *Weighted Averaging of Local Fittings* (*WALF*) and *Continuous Moving Frames* (*CMF*). These methods are both based on weighted least squares polynomial fittings and guarantee C^0 continuity. Unlike existing methods for reconstructing surfaces, our methods are applicable to surface meshes composed of triangles and/or quadrilaterals, can achieve third and even higher order accuracy, and have integrated treatments for sharp features. We present the theoretical framework of our methods, experimental comparisons against other methods, and its applications in a number of meshing operations.

Key words: mesh generation; curves and surfaces; mesh adaptivity; high-order methods; accuracy

1 Introduction

Surface meshes and their manipulations are critical for meshing, numerical simulations, and many other related problems. Some example problems that involve manipulating surface meshes include mesh generation and mesh enhancement for finite element or finite volume computations [5], mesh smoothing in ALE methods [1], and mesh adaptation in moving boundary problems [11]. In many of these problems, a continuous CAD model may not be available. Instead, only a surface mesh, typically with piecewise linear or bilinear faces, is available.

In this paper, we consider the problem of reconstructing a highly accurate, continuous geometric support from a given surface mesh. We refer to this problem as *high-order surface reconstruction* (or simply *high-order reconstruction*). Besides meshing, this reconstruction problem also arises in computer graphics [2] and geometric

* Corresponding author. Email: jiao@ams.sunysb.edu.

modeling [17]. In general, the high-order reconstruction should satisfy some (if not all) of the following requirements:

Continuity: The reconstructed surface should be continuous to some degree (e.g., C^0, C^1, or C^2 continuous, depending on applications).

Feature preservation: The reconstruction should preserve sharp features (such as ridges and corners) in the geometry.

Geometric accuracy: The reconstruction should be accurate and asymptotically convergent to the exact surface to certain order under mesh refinement.

Stability: The reconstruction should be numerically stable and must not be oscillatory under noise.

Note that different applications may have emphasis on different aspects of the problem. For example, in computer graphics and geometric design, the visual effect may be the ultimate goal, so smoothness and feature preservation may be most important. Therefore, methods proposed for such applications tend to focus on the first two issues, and the numerical issues of asymptotic convergence and stability are mostly ignored. Our focus in this paper is on meshing for finite element or finite volume computations, for which these numerical issues are very important. Indeed, the low order of accuracy of the geometry would necessarily limit the accuracy of the solutions of differential equations, and numerical instabilities and excessive oscillations can have even more devastating effect to numerical simulations.

In this paper, we present two methods, called *Weighted Averaging of Local Fittings* (*WALF*) and *Continuous Moving Frames* (*CMF*), for reconstructing a feature-preserving, high-order surface from a given surface mesh. Both methods are based on the assumptions that the vertices of the mesh accurately sample the surface, and the faces of the mesh correctly specify the topology of the surface, and utilize the numerical techniques of weighted least squares approximations and piecewise polynomial fittings. These methods apply to surface meshes composed of triangles and/or quadrilaterals, and also to curves (such as ridge curves on a surface). Unlike existing methods, which are typically only first or second order accurate, our methods can achieve third- and even higher order accuracy, while guaranteeing C^0 continuity. For its weighted least squares nature, these methods are also tolerant to noise. We present the theoretical framework of our methods. We also present experimental comparisons of our methods against some others, and its applications in a number of meshing operations.

The remainder of the paper is organized as follows. Section 2 presents some background knowledge, including local polynomial fittings and weighted least squares approximations. Section 3 describes the new methods for high-order surface reconstruction and compares them with some others. Section 4 applies our methods with a number of meshing operations for triangle and quadrilateral meshes and meshes with sharp features. Section 5 concludes the paper with a discussion.

2 Preliminaries: Vertex-Based Polynomial Fittings

Our high-order reconstruction is based on local polynomial fittings and weighted least squares approximations. We have successfully used these techniques previously to compute differential quantities of discrete surfaces (such as normals and curvatures) to high-order accuracy; see e.g. [13, 18]. However, due to their local nature, those approaches for computing differential quantities do not provide a continuous, global reconstruction of a surface. We hereafter briefly review these techniques and then adapt them to high-order reconstruction in the next section. For more details on the theoretical background, readers are referred to [13] and references therein.

2.1 Local Polynomial Fitting

Local polynomial fittings, also known as *Taylor polynomials* in numerical analysis [8], are based on the well-known Taylor series expansions about a point. We are primarily concerned with surfaces, so the local fitting is basically an interpolation or approximation to a neighborhood of a point P under a local parameterization (say, with parameters u and v), where P corresponds to $u = 0$ and $v = 0$. The polynomial fitting may be defined over the global xyz coordinate system or a local uvw coordinate system. In the former, the neighborhood of the surface is defined by the *coordinate function* $\boldsymbol{f}(u, v) = [x(u, v), y(u, v), z(u, v)]$. In the latter, assuming the uv-plane is approximately parallel with the tangent plane of the surface at P, each point in the neighborhood of the point can be transformed into a point $[u, v, f(u, v)]$ (by a simple translation and rotation), where f is known as the *local height function*.

Let $\boldsymbol{u}$ denote $[u, v]^T$. Let $\varphi(\boldsymbol{u})$ denote a smooth bivariate function, which may be the local height function or the x, y, or z component of the coordinate function for a parametric surface. Let c_{jk} be a shorthand for $\frac{\partial^{j+k}}{\partial u^j \partial v^k}\varphi(\mathbf{0})$. Let d be the desired degree of the polynomial fitting, and it is typically small, say ≤ 6. If $\varphi(\boldsymbol{u})$ has $d+1$ continuous derivatives, it can be approximated to $(d+1)$st order accuracy about the origin $\boldsymbol{u}_0 = [0, 0]^T$ by

$$\varphi(\boldsymbol{u}) = \underbrace{\sum_{p=0}^{d} \sum_{j,k\geq 0}^{j+k=p} c_{jk} \frac{u^j v^k}{j!k!}}_{\text{Taylor polynomial}} + \underbrace{\sum_{j,k\geq 0}^{j+k=d+1} \frac{\partial^{j+k}}{\partial u^j \partial v^k} \varphi(\tilde{u}, \tilde{v}) \frac{\tilde{u}^j \tilde{v}^k}{j!k!}}_{\text{remainder}}, \tag{1}$$

where $0 \leq \tilde{u} \leq u$ and $0 \leq \tilde{v} \leq v$.

Suppose we have a set of data points, say $[u_i, v_i, \varphi_i]^T$ for $i = 1, \ldots, m-1$, sampled from a neighborhood near P on the surface. Substituting each given point into (1), we obtain an approximate equation

$$\sum_{p=0}^{d} \sum_{j,k\geq 0}^{j+k=p} \left(\frac{u_i^j v_i^k}{j!k!} \right) c_{jk} \approx \varphi_i, \tag{2}$$

which has $n = (d+1)(d+2)/2$ unknowns (i.e., c_{jk} for $0 \leq j+k \leq d$, $j \geq 0$ and $k \geq 0$), resulting in an $m \times n$ rectangular linear system. Note that one could force the polynomial to pass through point P by setting $c_{00} = 0$ and removing its corresponding equation, reducing to an $(m-1) \times (n-1)$ rectangular linear system. This may be beneficial if the points are known to interpolate a smooth surface.

Let us denote the rectangular linear system obtained from (2) as

$$\boldsymbol{V}\boldsymbol{X} \approx \boldsymbol{F}, \tag{3}$$

where $\boldsymbol{X}$ is an n-vector composed of c_{jk}, and $\boldsymbol{V}$ is $m \times n$, known as a *generalized Vandermonde matrix*. For a local height function, $\boldsymbol{F}$ is an m-vector composed of f_i; for a parametric surface, $\boldsymbol{F}$ is an $m \times 3$ matrix, of which each column corresponds to a component of the coordinate function.

The above formulations can be easily adapted to curves in 2-D or 3-D, by using the univariable instead of the bivariable version of Taylor series expansions. For a curve in 3-D, the parameterization has only one parameter (say u), and the local height function has two components. When applied to a surface mesh, the point P is typically a vertex, and its neighborhood are typically some k-ring neighborhood. Following [13], we allow k to have half-ring increments:

- The *1-ring neighbor faces* of a vertex v are the faces incident on v, and the *1-ring neighbor vertices* are the vertices of these faces.
- The *1.5-ring neighbor faces* are the faces that **share an edge** with a 1-ring neighbor face, and the *1.5-ring neighbor vertices* are the vertices of these faces.
- For an integer $k \geq 1$, the $(k+1)$*-ring neighborhood* of a vertex is the union of the 1-ring neighbors of its k-ring neighbor vertices, and the $(k+1.5)$*-ring neighborhood* is the union of the 1.5-ring neighbors of the k-ring neighbor vertices.

Figure 1 illustrates these neighborhood definitions up to 2.5 rings. We typically choose k to be $(d+1)/2$ (for non-noisy surface) or $d/2+1$ (for noisy surface), but may also enlarge k if there are fewer than 1.5 times of the required number of points in the k-ring.

2.2 Weighted Least Squares Approximation

Numerically, (3) can be solved using the framework of *weighted linear least squares* [7, p. 265], i.e., to minimize a weighted norm (or semi-norm),

$$\min_{\boldsymbol{X}} \|\boldsymbol{V}\boldsymbol{X} - \boldsymbol{F}\|_{\boldsymbol{\Omega}} = \min_{\boldsymbol{X}} \|\boldsymbol{\Omega}(\boldsymbol{V}\boldsymbol{X} - \boldsymbol{F})\|_2, \tag{4}$$

where $\boldsymbol{\Omega}$ is a *weighting matrix*. Typically, $\boldsymbol{\Omega}$ is an $m \times m$ diagonal matrix, whose ith diagonal entry ω_i assigns a priority to the ith point $[u_i, v_i]^T$ by scaling the ith row

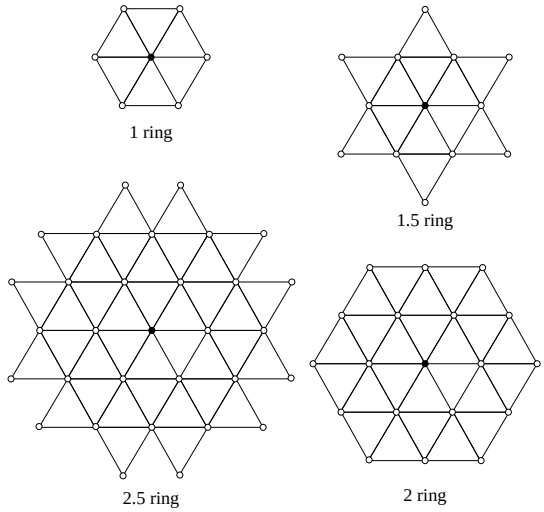

Fig. 1. Examples of 1-, 1.5-, 2-, and 2.5-ring vertices for typical vertex in triangle mesh. Each image depicts the neighborhood of the center black vertex.

of $\boldsymbol{V}$. It is desirable to assign lower priorities to points that are farther away from the origin or whose normals differ substantially from the w direction of the local coordinate frame, such as that defined in (8).

The formulation (4) is equivalent to the linear least squares problem

$$\tilde{\boldsymbol{V}}\boldsymbol{X} \approx \boldsymbol{B}, \text{ where } \tilde{\boldsymbol{V}} = \boldsymbol{\Omega}\boldsymbol{V} \text{ and } \boldsymbol{B} = \boldsymbol{\Omega}\boldsymbol{F}. \tag{5}$$

In general, $\tilde{\boldsymbol{V}}$ is $m \times n$ and $m \geq n$. A technical difficulty is that this linear system may be very ill-conditioned (i.e., the singular values of $\tilde{\boldsymbol{V}}$ may differ by orders of magnitude) due to a variety of reasons, such as poor scaling, insufficient number of points, or degenerate arrangements of points [14]. The conditioning number of $\tilde{\boldsymbol{V}}$ can be improved by using a scaling matrix $\boldsymbol{S}$ and changing the problem to

$$\min_{\boldsymbol{Y}} \|\boldsymbol{A}\boldsymbol{Y} - \boldsymbol{B}\|_2, \text{ where } \boldsymbol{A} = \tilde{\boldsymbol{V}}\boldsymbol{S} \text{ and } \boldsymbol{Y} = \boldsymbol{S}^{-1}\boldsymbol{X}. \tag{6}$$

We chose $\boldsymbol{S}$ to be a diagonal matrix. Let $\tilde{\boldsymbol{v}}_i$ denote the ith column of $\tilde{\boldsymbol{V}}$. The ith diagonal entry of $\boldsymbol{S}$ is chosen to be $\|\tilde{\boldsymbol{v}}_i\|_2$, which approximately minimizes the condition number of $\tilde{\boldsymbol{V}}\boldsymbol{S}$ [7, p. 265].

2.3 Accuracy and Stability of Least Squares Polynomial Fittings

The local least squares polynomial fitting provides us the theoretical foundation for high-order reconstruction of surfaces, established by the following proposition [13]:

Proposition 1. *Given a set of points* $[u_i, v_i, \tilde{f}_i]$ *that interpolate a smooth height function* f *or approximate* f *with an error of* $O(h^{d+1})$*. Assume the point distribution and the weighting matrix are independent of the mesh resolution, and the condition number of the scaled matrix* $\boldsymbol{A} = \tilde{\boldsymbol{A}}\boldsymbol{S}$ *in (6) is bounded by some constant. The degree-*d *weighted least squares fitting approximates* c_{jk} *to* $O(h^{d-j-k+1})$*.*

Here, h is a local measure of mesh resolution (such as average edge length of the k-ring neighborhood). We refer to readers to [13] for the proof of the proposition. Note that a necessary condition for the accuracy is that the condition number of the scaled matrix $\boldsymbol{A}$ must be bounded, but it is not automatically the case even if the number of points is greater than the number of unknown coefficients. We achieve well-conditioning by either expanding the neighborhood or reducing the degree of fitting if the condition number is determined to be large, and in turn guarantee both accuracy and stability.

3 Continuous, High-Order Surface Reconstruction

The method described in the previous section applies locally at each individual vertex of a mesh. There was no coordination among the local fittings at different vertices, so the method does not reconstruct a continuous surface. To construct a continuous surface, there are at least three different options:

1. compute multiple local fittings at vertices and then compute a weighted averaging of these fittings;
2. enforce continuity of local coordinate frames and weights for local fittings;
3. introduce additional control points to define continuous/smooth surface patches.

Most methods in the literature use the latter two options. For example, the moving least squares [15] uses the second option to construct a C^∞ surface from a point cloud. Walton's method [17] adopted by Yams [4, 3] uses the third option. In this section, we describe two methods that belong to the first two categories, respectively. We will first focus on triangle meshes for smooth surfaces in this section, and will present the extension to quadrilateral meshes and for meshes with sharp features in the next section when describing their applications in meshing.

3.1 Weighted Averaging of Local Fittings (WALF)

A simple approach to construct a high-order surface is to compute a weighted average of the local fittings at vertices. We refer to this approach as *Weighted Averaging of Local Fittings* (*WALF*). To achieve continuity of the surface, the weights used by the weighted averaging must be continuous over the mesh. One such a choice is

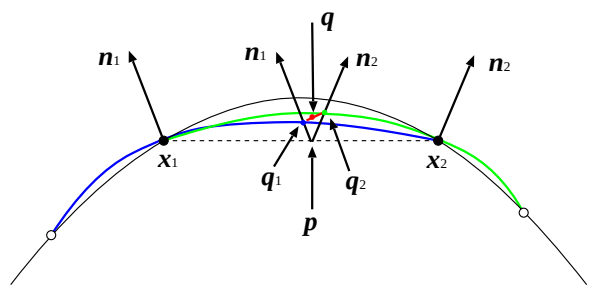

Fig. 2. 2-D illustration of weighted averaging of local fitting. The black curve indicates the exact curve. The blue and green curves indicate the fittings at vertices $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively. $\boldsymbol{q}$ is the WALF approximation of point $\boldsymbol{p}$ and is computed as a weighted average of the points $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ on the blue and green curves, respectively.

the barycentric coordinates of the vertices over each triangle. Consider a triangle composed of vertices $\boldsymbol{x}_i$, $i = 1, 2, 3$, and any point $\boldsymbol{p}$ in the triangle. For each vertex $\boldsymbol{x}_i$, we obtain a point $\boldsymbol{q}_i$ for $\boldsymbol{p}$ from the local fitting in the local uvw coordinate frame at $\boldsymbol{x}_i$, by projecting $\boldsymbol{p}$ onto its uv-plane. Let ξ_i, $i = 1, 2, 3$ denote the barycentric coordinates of $\boldsymbol{p}$ within the triangle, with $\xi_i \in [0, 1]$ and $\sum_{i=1}^{3} \xi_i = 1$. We define

$$\boldsymbol{q}(\boldsymbol{u}) = \sum_{i=1}^{3} \xi_i \boldsymbol{q}_i(\boldsymbol{u}) \tag{7}$$

as the approximation to point $\boldsymbol{p}$. Figure 2 shows a 2-D illustration of this approach, where ξ_i are the barycentric coordinates of point $\boldsymbol{p}$ within the edge $\boldsymbol{x}_1\boldsymbol{x}_2$.

WALF constructs a C^0 continuous surface, as can be shown using the properties of finite-element basis functions: The barycentric coordinates at each vertex of a triangle corresponds to the shape function of the vertex within the triangle, and the shape function of the vertex in all elements forms a C^0 continuous basis function (i.e., the linear pyramid function for surfaces or the hat function for curves). Let ϕ_i denote the basis function associated with the ith vertex of the mesh, and it is zero almost everywhere except within the triangles incident on the ith vertex. Therefore, $\boldsymbol{q}$ can be considered as a weighted average of the polynomials at all the vertices,

$$\boldsymbol{q}(\boldsymbol{u}) = \sum_{i=1}^{n} \phi_i(\boldsymbol{u}) \boldsymbol{q}_i(\boldsymbol{u}),$$

and then it is obvious that $\boldsymbol{q}$ is C^∞ within each triangle and C^0 over the whole mesh.

The idea of WALF is intuitive, but the analysis of its accuracy is by no means straightforward. If the coordinate systems were the same at all vertices, then the analysis would have been easy, as $\boldsymbol{q}$ would have inherited the accuracy of $\boldsymbol{q}_i$. However in our case, the local fittings at the three vertices of a triangle are in different coordinate systems in general, and this discrepancy of coordinate systems can lead to additional

error terms. Under the same assumptions as Proposition 1, we obtain the following property of WALF.

Proposition 2. *Given a mesh whose vertices approximate a smooth surface Γ with an error of $O(h^{d+1})$, the distance between each point on the WALF reconstructed surface and its closest point on Γ is $O(h^{d+1} + h^6)$.*

Note that the above proposition gives an upper bound of the error, so the lower bound of the convergence rate is $\min(6, d+1)$. The bound of h^6 is due to the discrepancy of local coordinate systems at different vertices. The proof is given in the Appendix.

3.2 Continuous Moving Frames (CMF)

WALF is a simple and intuitive method, but its order of accuracy may be limited. We now present a method that can overcome this limitation by using local coordinate frames that move continuously from point to point. We refer to such a scheme as *continuous moving frame* (*CMF*). The basic idea is to use the finite-element basis functions to construct continuous moving frames and weights for local fittings. In particular, assume each vertex has an approximate normal direction at input. Consider a triangle $\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$ and any point $\boldsymbol{p}$ in the triangle. Let $\hat{\boldsymbol{n}}_i$ denote the unit vertex normal at the ith vertex. We compute a normal at $\boldsymbol{p}$ as

$$\hat{\boldsymbol{n}} = \sum_{i=1}^{3} \xi_i \hat{\boldsymbol{n}}_i \Bigg/ \left\| \sum_{i=1}^{3} \xi_i \hat{\boldsymbol{n}}_i \right\| .$$

Given $\hat{\boldsymbol{n}}$, we construct a local uvw coordinate system along axes $\hat{\boldsymbol{s}}$, $\hat{\boldsymbol{t}}$, and $\hat{\boldsymbol{n}}$, where $\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{t}}$ form an orthonormal basis of the tangent plane. Within this local coordinate frame, we formulate the weighted least squares as

$$\|\boldsymbol{\Omega}\boldsymbol{V}\boldsymbol{X} - \boldsymbol{\Omega}\boldsymbol{F}\|_2,$$

where $\boldsymbol{V}$ again is the generalized Vandermonde matrix, and $\boldsymbol{\Omega}$ is the weight matrix.

In practice, the Vandermonde matrix for a point $\boldsymbol{p}$ should involve a small stencil in the neighborhood of the triangle. We use the union of the stencils of the three vertices of the triangle. Conceptually, it is helpful to consider the Vandermonde matrix involving all the points of the mesh, but the weight matrix $\boldsymbol{\Omega}$ assigns a zero weight for each point that is not in the stencil. For the reconstructed surface to be smooth, it is important that $\boldsymbol{\Omega}$ is continuous as the point $\boldsymbol{p}$ moves within the geometric support of the mesh. In addition, it is also important that $\boldsymbol{\Omega}$ is invariant of rotation of tangent plane (i.e., be independent of the choice of $\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{t}}$).

We define the weight as follows. For $\boldsymbol{p}$ within the triangle $\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$, we first define a weight for each vertex (say jth vertex) in the mesh corresponding to $\boldsymbol{x}_i$ as

$$w_{ij} = \begin{cases} \xi_i \left(\hat{\boldsymbol{n}}_i^T \hat{\boldsymbol{n}}_j\right)^+ e^{-\|\boldsymbol{x}_j - \boldsymbol{p}\|^2/h_i^2} & \text{if vertex } j \text{ is in stencil of } i\text{th vertex} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where

$$\left(\hat{\boldsymbol{n}}_i^T \hat{\boldsymbol{n}}_j\right)^+ = \begin{cases} \hat{\boldsymbol{n}}_i^T \hat{\boldsymbol{n}}_j & \text{if } \hat{\boldsymbol{n}}_i^T \hat{\boldsymbol{n}}_j \geq \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

for some small $\epsilon \geq 0$ and h_i is a local mesh-resolution measure at the ith vertex. Then for the weighting matrix, the weight for jth vertex is then $\sum_{i=1}^{3} w_{ij}$. In the actual implementation, for simplicity we list the jth vertex separately for its appearance in the stencil of each vertex of the triangle, and include only the vertices whose weights are nonzeros in $\boldsymbol{V}$ and $\boldsymbol{\Omega}$.

Similar to WALF, CMF constructs a C^0 continuous surface, because $\boldsymbol{\Omega}$, $\boldsymbol{V}$, and $\boldsymbol{F}$ are all C^0 continuous, as long as the resulting linear system is well-conditioned. The accuracy of CMF follows that for weighted least squares approximation in [13], and we obtain the following properties of CMF.

Proposition 3. *Given a mesh whose vertices approximate a smooth surface Γ with an error of $O(h^{d+1})$, the shortest distance from each point on the CMF reconstructed surface to Γ is $O(h^{d+1})$.*

Relationship with Moving Least Squares. The idea of using moving frames is not new, and goes back to Élie Cartan for differential geometry. One of the incarnations of the idea of using moving frames for discrete surfaces is the so-called *moving least squares* (*MLS*) for point clouds [15]. CMF shares some similarities to MLS. In particular, they are both based on weighted least squares approximations within some local frames. However, they also differ in some fundamental ways. First, moving least squares uses global weighting functions that are exponential in the distance, and theoretically, MLS is C^∞. However, because global weighting functions are too expensive to compute, practical implementations typically truncate small weights to zeros, leading to a loss of continuity. In contrast, CMF uses only a local support by construction. Second, MLS does not guarantee the order of accuracy, because its weights are global and purely based on Euclidean distance. Although its convergence was conjectured in [15], we have observed that MLS does not converge even for simple geometries such as a torus. In contrast, CMF uses the mesh connectivity as a clue in selecting the stencils, instead of based on Euclidean distance. Third, CMF can take into account the normals in the weighting function, to filter out points across sharp features. This allows CMF to handle surfaces with sharp features in a natural way, which is important for meshing operations. On the other hand, it is difficult to treat sharp features in the framework of MLS. Because of their local supports, CMF is more easily adapted to treat sharp features, as we describe in the next section.

3.3 Experimental Results

We report some experimental results of our two proposed methods, and compare them with some other methods. We first show the mesh convergence study of WALF

and CMF. While it is typically unnecessary to use schemes with higher than third or fourth order accuracy, to demonstrate the capabilities and limitations of these two methods, we report results with polynomials of up to degree 6. We performed our experiment using a torus with in-radius of 0.7 and outer-radius of 1.3, with an unstructured triangular mesh. We considered three levels of mesh refinement. The coarsest mesh has 329 vertices and 658 triangles, whereas the finest mesh has 21,156 vertices and 42,312 triangles. In this test, we randomly generated 10 points on each face of the mesh, then project them onto high order surface using WALF or CMF. We compute the error as the shortest distance from each approximate point to the torus.

Figure 3 shows the L_∞ errors of WALF and CMF for the meshes. In the figure, the horizontal axis corresponds to the level of mesh refinement, and the vertical axis corresponds to the L_∞ errors. In the legends, the "degree" indicates the degree of polynomial fittings used, and "linear" indicates the error for linear interpolation within triangles. We show the average convergence rates along the right of the plots for each curve, which was calculated as $\log\left(\text{error}_3/\text{error}_{\text{base}}\right) / \log\left(h_3/h_{\text{base}}\right)$, where error_i denotes the L_∞ error of all the randomly inserted points for the ith coarsest mesh, and h_i is the maximum edge length of the corresponding mesh. We chose the base to be 0 for CMF and 1 for WALF, because the errors for WALF were large for degrees 5 and 6 for the coarsest mesh, leading to artificially too large convergence rates.

From the figures, it is obvious that quadratic and higher-degree fittings produced far more accurate results than linear interpolation. Both WALF and CMF achieved a convergence rate of $(d+1)$ when d is odd and higher than $(d+2)$ when d is even. The superconvergence for even-degree fittings is likely due to statistical error cancellations of the leading error terms, which are of odd degrees. However, such error cancellations are not guaranteed when the points are very far from being symmetric, especially near boundaries or sharp features.

Some conclusions can be drawn from our comparisons between WALF with CMF. In terms of accuracy, we note that WALF gave smaller errors (up to 50% smaller) than CMF for finer meshes, although they delivered very similar convergence rates. The reason for the smaller errors for WALF was probably that WALF uses a smaller

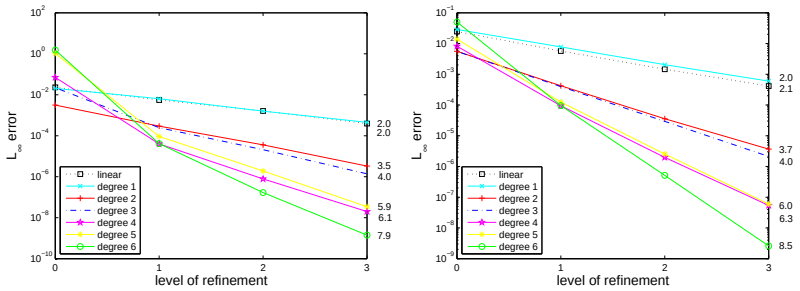

Fig. 3. L_∞ errors of WALF (left) and CMF (right) under mesh convergence for torus. Both WALF and CMF achieve $(d+1)$st or higher order accuracy for degree-d polynomials.

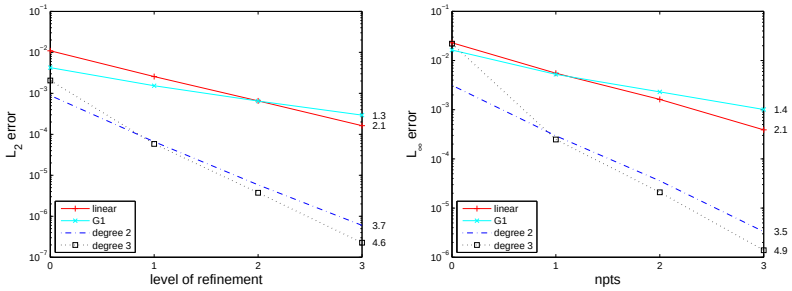

Fig. 4. Comparison of errors using linear interpolation, Walton's method (labeled as G1), and WALF using quadratic and cubic fittings.

stencil for each polynomial fitting. In terms of efficiency, WALF and CMF are comparable when approximating a single point. However, when approximating many points, WALF allows reusing the polynomial fittings computed at the vertices, and hence it can have smaller amortized cost. Furthermore, WALF is also more versatile, because one can change the polynomial at each vertex in WALF (such as adapting its degree of polynomial) without losing C^0 continuity, but the same cannot be done with CMF. Therefore, we favor WALF over CMF, unless some application requires guaranteed seventh or higher order accuracy.

Besides WALF and CMF, some other methods have been developed for high order reconstructions and been used in the meshing community. One method that is worth noting is that proposed by Walton [17] and adopted by Frey for surface meshing [3]. One property of Walton's method is that it achieves C^1 (or G^1) continuity for the reconstructed mesh. However, there does not seem to be any analysis of the accuracy of Walton's method in the literature. Figure 4 shows a comparison of the errors of Walton's method as well as WALF using quadratic and cubic fittings, with linear interpolation as the baseline of comparison. The two figures show the errors in L_∞ and L_2 norms for the torus. From the result, it is evident that Walton's method converges only linearly and is actually less accurate than linear interpolation for finer meshes. Our preliminary theoretical analysis also suggests that Walton's method can be at most second order accurate (i.e., no better than linear interpolation), and its practical accuracy may be worse because its dependence on the tangent directions of the edges, which are in general unavailable and can be estimated only to low accuracy. Therefore, it is obvious that C^1 (and in fact even C^∞) continuity does not imply accuracy of the reconstruction, although they may produce smooth looking. On the other hand, as we will show in the next section, high-order methods with C^0 continuity typically produce errors that are too small to cause any noticeable discontinuities.

4 Applications to Meshing and Finite Elements

The targeted applications for high-order surface reconstruction for this paper are meshing for finite element analysis. We hereafter further customize our framework for meshing and then apply the resulting techniques to meshing operations.

4.1 Quadrilateral Elements and Sharp Features

To be generally applicable to meshing, the reconstruction techniques should work not only for triangulations of smooth surfaces but also for meshes with quadrilateral elements (including quadrilateral meshes or quad-dominant meshes) and surfaces with sharp features.

Generalization to Meshes Containing Quadrilateral. For a quadrilateral element, we need to use the finite element shape functions N_i. Let ξ and η be the two natural coordinates within a quadrilateral element, with $0 \leq \xi \leq 1$ and $0 \leq \eta \leq 1$, then the shape functions associated with the four vertices are

$$N_1 = (1-\xi)(1-\eta), \quad N_2 = \xi(1-\eta),$$
$$N_3 = (1-\xi)\eta, \quad N_4 = \xi\eta.$$

A key issue is the selection of stencils for a quadrilateral mesh or quad-dominant mesh. The definition of n-ring neighbors in [13] tends to produce too many points for quadrilateral meshes. We redefine the neighborhood of a vertex as follows:

- The *0-ring* of a vertex is the vertex itself ;
- The k-*ring* vertices of a vertex (where $k = 1, 2, 3, \ldots$) is the set of vertices that are share an edge with a vertex in the (k-1)-ring;
- The $(k + 0.5)$-ring of a vertex (where $k = 1, 2, 3, \ldots$) is the union of k-ring vertices and the vertices that share elements with an edge between two vertices in the k-ring.

For a triangle mesh, the above definition is equivalent to that in [13]. However, this definition is also well-suited to other types of surface meshes. Figure 5 shows the 1-, 1.5-, 2-, and 2.5-rings of a typical vertex in a quadrilateral mesh or a quad-dominant mesh. In general for degree-d fittings, we find it most effective to use a ring of $(d + 1)/2$ for a mesh without noise or a ring of $d/2 + 1$ or larger for meshes with noise.

Treatment of Sharp Features. Sharp features, such as ridges and corners, are challenging problems in their own right. We have so far implemented a simple treatment for sharp features. First, we identify feature edges and vertices and connect the feature edges to form ridge curves using an algorithm such as that in [9, 10]. We treat the ridge edges as internal boundaries within the mesh and require the k-ring neighbors

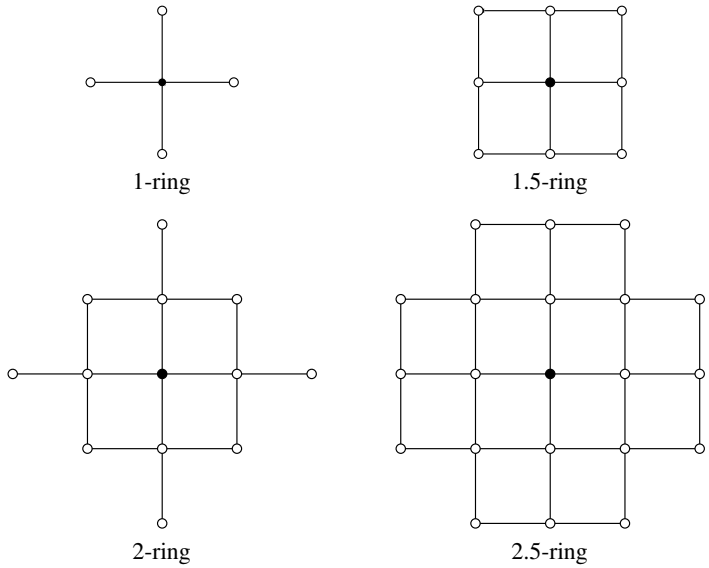

Fig. 5. Examples of 1-, 1.5-, 2-, and 2.5-rings of typical vertex in quadrilateral mesh. Each image depicts the neighborhood of the center black vertex.

of vertices do not go across ridge curves. This is accomplished by virtually splitting the mesh along ridge curves in our mesh data structure. For ridge curves themselves, we separate them into sub-curves that do not have corners in their interior. For each sub-curve, we perform a high-order reconstruction using either WALF or CMF for curves. This treatment is sufficient for most meshing operations.

4.2 High-Order Finite Elements

An application of our method is to construct a high order (in particular, quadratic or cubic) finite element mesh from a given mesh with only linear elements and accurate vertex coordinates. This problem has practical relevance, because some mesh generators produce only a mesh with linear elements from an accurate CAD model, and it may be desirable to reconstruct high-order elements without having to access the CAD model. We formally stated the problem as follows: Given a mesh with linear elements, assume the vertices are sufficiently accurate (e.g., they are exact or are at least third or fourth-order accurate), construct a finite element mesh with quadratic or cubic elements with third or fourth order accuracy.

When using high-order surface reconstruction, this problem can be solved in the following procedure:

1. For each element σ, loop through its edges. If there is not an element that is abutting σ and has an ID smaller than that of σ, assign a new node ID to each

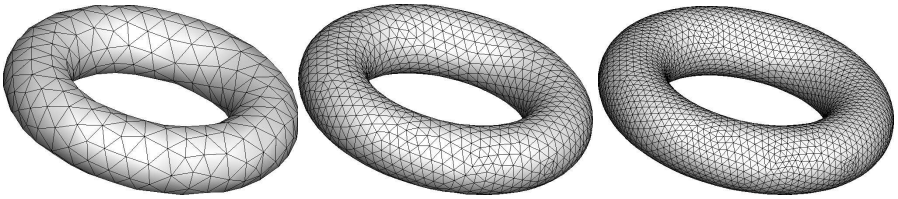

Fig. 6. Illustration of generating high-order finite elements from given mesh. Left: a coarse torus with linear elements; Middle: same mesh but with quadratic elements, visualized by decomposing each triangle into four triangles. Right: same mesh but with cubic elements, visualized by decomposing each triangle into nine triangles.

node on the edge; if a node ID has already been assigned in the adjacent element, retrieve the node ID from that adjacent element;

2. Loop through all elements to assign new nodes IDs for new nodes on faces;
3. Expand the array for nodal coordinates, and evaluate the position for each new vertex using high-order surface reconstruction.

We have implemented this procedure for reconstructing quadratic and cubic elements from linear triangles or bilinear quadrilaterals using WALF and CMF. Figure 6 shows an example for meshes generated with quadratic and cubic elements for a torus. The high-order schemes produced notable improvements to the smoothness of the surface to linear approach, and the overall errors are significantly smaller. Note that in actual numerical simulations, not only the geometry but also some field variables defined on the mesh need to be reconstructed to high order. The same high-order reconstruction we presented can be used for that purpose, but it is beyond the scope of this paper.

4.3 Uniform Mesh Refinement

A problem related to generating a high-order mesh is a uniform refinement of a surface mesh. The problem may be stated as follows: given a coarse surface mesh with sufficiently accurate vertex coordinates, construct a finer surface mesh by subdividing the elements. Like the previous problem, uniform mesh refinement introduces additional nodes to edges and/or faces, but in addition it also introduces new edges to subdivide the elements. Figure 7 shows an example of refining a quadrilateral mesh with sharp features, and Figure 8 shows that for a triangular mesh of a dragon head. Note that if the new points are added onto the linear edges and faces, the refined mesh is not only inaccurate but also nonsmooth, as evident in the left image of the figures. This problem is resolved by using high order reconstructions. The right image of Figure 7 and bottom-right image of Figure 8 show a refinement mesh using WALF and feature treatments. The resulting meshes are much smoother and more accurate. This procedure can be useful for generating high-quality finer resolution meshes from a mesh without requiring access to the CAD model.

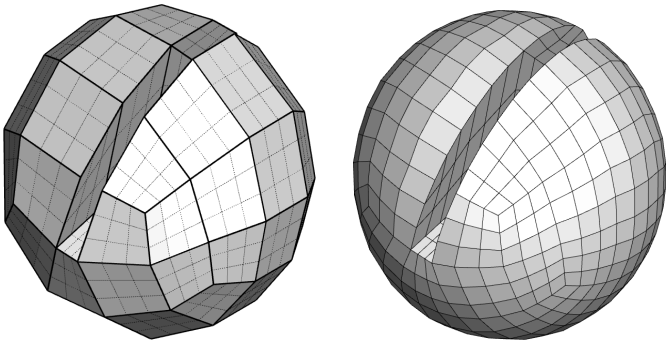

Fig. 7. Example of refining quadrilateral mesh by subdividing each element into nine quadrilaterals. In left, dark lines show original coarse mesh, and dashed lines show linear subdivision. Right images shows refined mesh using cubic fitting with WALF and feature treatments.

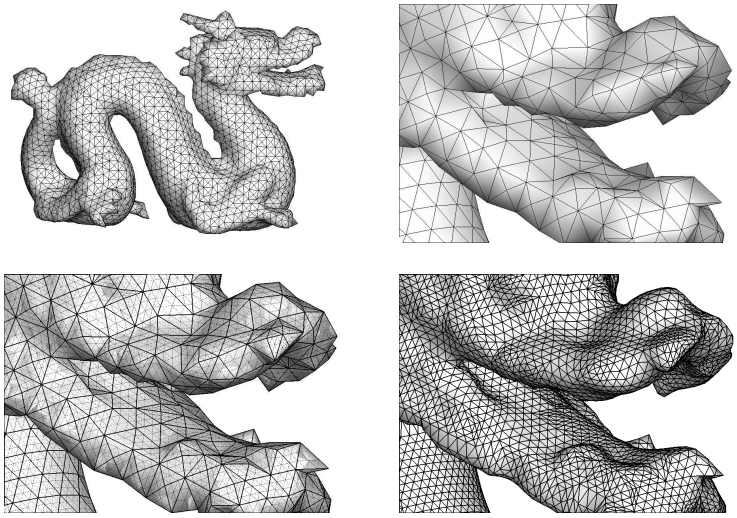

Fig. 8. Example of refining triangular mesh by subdividing each element into four triangles. Upper row shows a dragon mesh and the zoom in near the head. Lower left image shows a refined mesh using linear interpolation. Lower right image shows refined mesh using WALF with quadratic fitting and feature treatments.

4.4 Mesh Smoothing and Mesh Adaptivity

More general meshing applications of our techniques are the smoothing and adaptivity of surface meshes. In these settings, not only new vertices may be added, existing vertices may also be moved. For these meshing operations, a common approach is to keep the original mesh during mesh smoothing/adaptation, and project new vertices onto the faceted, piecewise linear geometries (see e.g., [6]). Such an approach has only second order accuracy. Another approach was taken by Frey [3], who constructed a G^1 continuous surface using Walton's method [17]. However, our exper-

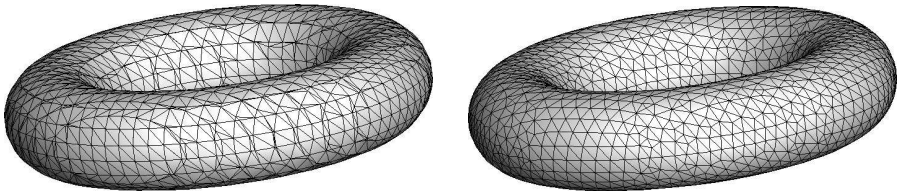

Fig. 9. Example of applying high-order reconstruction in meshing smoothing. The left image shows the original torus mesh, and the right image shows the smoothed mesh.

iments have shown that Walton's method is only about first order accurate, despite its G^1 continuity. Other methods have been developed (such as [16]), but none could deliver high order accuracy.

Instead of using low-order methods, we propose to use high-order surface reconstructions. As an example, we integrate high-order surface reconstruction with the variational mesh smoothing framework described in [12]. To utilize the high-order surface, we first compute the motion of each point within the tangent plane, and then project the new point onto the high-order surface. Note that the use of the tangent plane is useful, because it introduces an error that is high order compared to the tangential displacement, so the projection onto the high-order surface involves only a high order adjustment. Figure 9 shows an example of a smoothed surface mesh using WALF method compared to the original mesh generated by the isosurface function in MATLAB. The maximum angles were 166.1 and 128.8 degrees before and after smoothing, respectively, and minimum angles were 0.65 and 23.8 degrees, respectively. The smoothing process significantly improved the mesh quality while preserving the geometry to high order.

5 Conclusions and Discussions

In this paper, we studied the problem of reconstructing a high-order surface from surface meshes, in the context of meshing for finite element computations. We presented two methods, namely WALF and CMF, which are based on weighted least squares approximations with piecewise polynomial fittings. Unlike the traditional methods used in meshing, these methods can deliver very high order accuracy, independent of the order of the element of the mesh. The end results are high-order reconstructions that are efficient, noise resistant, feature preserving, and well suited for meshing and finite element computations. We demonstrate the high accuracy of our method compared to some other methods, and discussed its application in the context of reconstructing high-order finite elements, mesh refinement, mesh smoothing, and mesh adaptivity. Between WALF and CMF, we favor WALF for its simplicity and versatility (for example, WALF can be interpolatory, but CMF cannot be without sacrificing numerical stability), unless one requires guaranteed seventh or even higher order accuracy.

A property of our methods is that it enforces only C^0 continuity. Imposing only C^0 continuity allows us more freedom to achieve higher order accuracy. However, C^0 continuity may be deemed as a limitation of our method in some applications in computer-aided design and geometric modeling, especially when very coarse models with very few control points are used. In such cases, the loss of C^1 or C^2 continuity may lead to noticeable artifacts. We will investigate the reconstruction of C^1 surfaces with high-order accuracy in our future research.

Acknowledgement. This work was supported by National Science Foundation under award number DMS-0809285. The first author is also supported by DOE NEUP program under contract #DE-AC07-05ID14517 and by DoD-ARO under contract #W911NF0910306.

References

1. J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodriguez-Ferran. Arbitrary Lagrangian-Eulerian methods. In E. Stein, R. de Borst, and T. J. Hughes, editors, *Encyclopedia of Computational Mechanics*, chapter 14. Wiley, 2004.
2. S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics (TOG)*, 24(3), 2005.
3. P. J. Frey. About surface remeshing. In *Proc. 9th Int. Meshing Roundtable*, pages 123–136, Oct. 2000.
4. P. J. Frey. Yams: A fully automatic adaptive isotropic surface remeshing procedure. Technical report, INRIA, 2001. RT-0252.
5. P. J. Frey and P. L. George. *Mesh Generation: Application to finite elements*. Hermes, 2000.
6. R. Garimella. Triangular and quadrilateral surface mesh quality optimization using local parametrization. *Computer Methods in Applied Mechanics and Engineering*, 193(9-11):913–928, 2004.
7. G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins, 3rd edition, 1996.
8. M. T. Heath. *Scientific Computing: An Introductory Survey*. McGraw-Hill, New York, 2nd edition, 2002.
9. X. Jiao. Volume and feature preservation in surface mesh optimization. In *Proc. 15th Int. Meshing Roundtable*, 2006.
10. X. Jiao and N. Bayyana. Identification of C^1 and C^2 discontinuities for surface meshes in CAD. *Comput. Aid. Des.*, 40:160–175, 2008.
11. X. Jiao, A. Colombi, X. Ni, and J. Hart. Anisotropic mesh adaptation for evolving triangulated surfaces. *Engineering with Computers*, 26:363–376, 2010.
12. X. Jiao, D. Wang, and H. Zha. Simple and effective variational optimization of surface and volume triangulations. In *Proc. 17th Int. Meshing Roundtable*, pages 315–332, 2008.
13. X. Jiao and H. Zha. Consistent computation of first- and second-order differential quantities for surface meshes. In *ACM Solid and Physical Modeling Symposium*, pages 159–170. ACM, 2008.
14. P. Lancaster and K. Salkauskas. *Curve and Surface Fitting: An Introduction*. Academic Press, 1986.

15. D. Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67:1517–1531, 1998.
16. I. B. Semenova, V. V. Savchenko, and I. Hagiwara. Two techniques to improve mesh quality and preserve surface characteristics. In *Proc. 13th Int. Meshing Roundtable*, pages 277–288, 2004.
17. D. Walton. A triangular G1 patch from boundary curves. *Computer-Aided Design*, 28(2):113–123, 1996.
18. D. Wang, B. Clark, and X. Jiao. An analysis and comparison of parameterization-based computation of differential quantities for discrete surfaces. *Computer Aided Geometric Design*, 26(5):510–527, 2009.

Appendix: Proof of Proposition 2

We analyze the accuracy for triangles, but it helps to refer to Figure 2 for a 2-D illustration. Let $\boldsymbol{q}_i^*$ denote the intersection of the exact surface with the direction $\boldsymbol{n}_i$ from a point $\boldsymbol{p}$ (i.e., $\boldsymbol{q}_i^*$ is the exact solution for $\boldsymbol{q}_i$ in the fitting at vertex $\boldsymbol{x}_i$). Let $\bar{\boldsymbol{q}}$ denote the closest point to point $\boldsymbol{q} = \sum_{i=1}^{3} \xi_i \boldsymbol{q}_i$ on the exact surface. Let $\boldsymbol{q}^* = \sum_{i=1}^{3} \xi_i \boldsymbol{q}_i^*$ and $\bar{\boldsymbol{q}}^*$ be its closest point on the surface. Then,

$$\|\boldsymbol{q} - \bar{\boldsymbol{q}}\| \leq \|\boldsymbol{q} - \bar{\boldsymbol{q}}^*\| \leq \|\boldsymbol{q} - \boldsymbol{q}^*\| + \|\boldsymbol{q}^* - \bar{\boldsymbol{q}}^*\|. \tag{10}$$

For $\|\boldsymbol{q} - \boldsymbol{q}^*\|$, note that$\|\boldsymbol{q} - \boldsymbol{q}^*\| \leq \sum_{i=1}^{3} \xi_i \|\boldsymbol{q}_i - \boldsymbol{q}_i^*\|$. When dth degree fittings are used, $\|\boldsymbol{q}_i - \boldsymbol{q}_i^*\| = O(h^{d+1})$, so

$$\|\boldsymbol{q} - \boldsymbol{q}^*\| = O(h^{d+1}). \tag{11}$$

For $\|\boldsymbol{q}^* - \bar{\boldsymbol{q}}^*\|$, $\|\boldsymbol{q}_1^* - \boldsymbol{q}_2^*\| = |\cos\theta_1| \|\boldsymbol{q}_1^* - \boldsymbol{p}\| + |\cos\theta_2| \|\boldsymbol{q}_2^* - \boldsymbol{p}\|$, where θ_i is the angle between $\boldsymbol{q}_1^* \boldsymbol{q}_2^*$ and $\hat{\boldsymbol{n}}_i$. Note that $\cos\theta_i = O(h)$, since by assumption $\hat{\boldsymbol{n}}_i$ is at least a first order approximation to the normal at $\boldsymbol{x}_i$, and a first order approximation to the normals at $\boldsymbol{q}_1^*$ and $\boldsymbol{q}_2^*$, whereas the line segment $\boldsymbol{q}_1^* \boldsymbol{q}_2^*$ is at least a first order approximation to a tangent direction at $\boldsymbol{q}_1^*$ and $\boldsymbol{q}_2^*$. Because $\boldsymbol{p}$ is a point on triangle $\boldsymbol{x}_1 \boldsymbol{x}_2 \boldsymbol{x}_3$, whose edge length is $O(h)$ by assumption, $\|\boldsymbol{p} - \boldsymbol{q}_i^*\| = O(h^2)$. Therefore,$\|\boldsymbol{q}_1^* - \boldsymbol{q}_2^*\| = O(h^3)$, and similarly for $\|\boldsymbol{q}_1^* - \boldsymbol{q}_3^*\|$ and $\|\boldsymbol{q}_2^* - \boldsymbol{q}_3^*\|$. Because $\boldsymbol{q}^*$ is a point on triangle $\boldsymbol{q}_1^* \boldsymbol{q}_2^* \boldsymbol{q}_3^*$,

$$\|\boldsymbol{q}^* - \bar{\boldsymbol{q}}^*\| = O(h^3)^2 = O(h^6). \tag{12}$$

Combining (10-12), we conclude that $\|\boldsymbol{q} - \bar{\boldsymbol{q}}\| = O(h^{d+1}) + O(h^6) = O(h^{d+1} + h^6)$.