A Comparison of Gradient Estimation Methods for Volume Rendering on Unstructured Meshes

Carlos D. Correa, *Member, IEEE,* Robert Hero, and Kwan-Liu Ma, *Senior Member, IEEE*

Abstract—This paper presents a study of gradient estimation methods for rendering unstructured-mesh volume data. Gradient estimation is necessary for rendering shaded isosurfaces and specular highlights, which provide important cues for shape and depth. Gradient estimation has been widely studied and deployed for regular-grid volume data to achieve local illumination effects, but has been otherwise for unstructured-mesh data. As a result, most of the unstructured-mesh volume visualizations made so far were unlit. In this paper, we present a comprehensive study of gradient estimation methods for unstructured meshes with respect to their cost and performance. Through a number of benchmarks, we discuss the effects of mesh quality and scalar function complexity in the accuracy of the reconstruction, and their impact in lighting-enabled volume rendering. Based on our study, we also propose two heuristic improvements to the gradient reconstruction process. The first heuristic improves the rendering quality with a hybrid algorithm that combines the results of the multiple reconstruction methods, based on the properties of a given mesh. The second heuristic improves the efficiency of its GPU implementation, by restricting the computation of the gradient on a fixed-size local neighborhood.

Index Terms—Volume rendering, Gradient estimation, Unstructured meshes

---❖---

1 INTRODUCTION

LIGHTING plays an important role in volume rendering. On one hand, shading and specular reflections provide important cues of shape and depth. On the other hand, diffuse shading along the contours of an isosurface helps disambiguate the overlapping structures that are common in semi-transparent rendering. For example, Fig. 1 shows the result of applying local illumination to two volumes sampled in unstructured meshes. On the left, lighting helps discover turbulent patterns that are lost in the unlit image. On the right, lighting helps elucidate the spatial relationships between the occluding isosurfaces. Without it, isosurfaces appear flat with no apparent depth disambiguation.

To properly apply local illumination to a 3D volume, we must estimate the gradient of the volume accurately at every single point, while hiding the effects of mesh resolution, which introduce undesired artifacts. Gradient estimation is well known and understood for regular grids, and its application is now part of commodity visualization systems. Due to the structured nature of regular grids, estimating the gradient is a rather simple task. The partial derivatives of a function with respect to the X,Y and Z dimensions are easily approximated using finite differences given the alignment of a voxel neighborhood with each of the axes. Unstructured meshes do not provide the same advantage. Usually based on finite element methods, these grids are used to discretize a scalar or vector field within a closed volume using a variety of cell types, such as tetrahedra, hexahedra and prisms. The use of cells of varying shape and size enables a better fit of the grid with complex geometries and adaptive refinement in regions of interest. Unstructured meshes, with the exception of an unstructured cloud of points, contain connectivity information that may be used to compute the gradient. However, it is not as straightforward as in regular grids. First, connected vertices do not align with the main axes, suggesting a variable contribution to each component of the gradient. Second, unstructured meshes usually contain elements of varying shape, where one dimension are better sampled than the others. The traversal of this connectivity is somewhat costly when compared to the convolution step usually required for structured grids. Even with no connectivity, computing a stencil around a given point demands a spatial search that may vary in size. For this reason, local illumination of unstructured-mesh volume data has been largely ignored. A simple mechanism would be to impose a regular grid and re-sample the volume accordingly, but it results in sampling problems. Re-sampling an unstructured mesh into a 3D regular grid at the Nyquist rate might result in very large volumes that exceed the available system or graphics memory.

In this paper, we present a comparison and quantitative analysis of the most prominent methods for linear gradient reconstruction for the purpose of lighting. Although higher order elements are becoming increasingly available, linear elements are still the most common representation for unstructured meshes, particularly for hardware accelerated rendering systems. The cost of storing and rendering higher order elements has not made it possible to render them at interactive rates in current graphics processing units. Understanding the factors that affect the quality of volume rendered images in linear unstructured meshes not only improves current visualization systems, but also paves the way for similar studies of higher order meshes.

We make the following contributions: (1) We present a comprehensive comparison of linear gradient reconstruction methods on unstructured meshes. Our systematic approach decomposes the accuracy of the gradient as the product of different factors, such as mesh resolution and element shape, and complexity of the scalar field. Our experiments suggest simple guidelines for applying the appropriate method for a given unstructured mesh to produce high-quality volume rendered images. To the best knowledge of the authors, this is the first attempt to obtain a comparison of the different gradient estimation methods from the visualization standpoint. (2) Based on our results, we present two heuristic improvements.

• *The authors are with the Visualization and Interface Design Innovation (VIDI) Research Group, Department of Computer Science, University of California, 2063 Kemper Hall, One Shields Avenue, Davis, CA 95616. E-mail: correac@cs.ucdavis.edu, rghero@ucdavis.edu, ma@cs.ucdavis.edu*
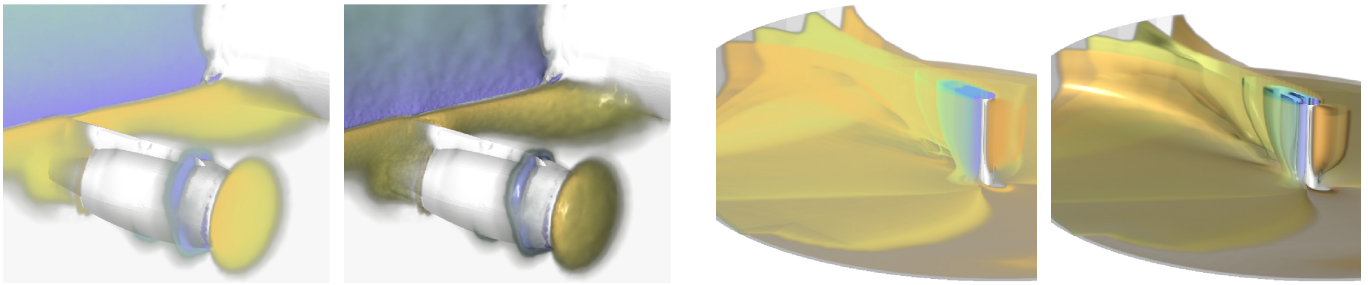
Fig. 1. Effect of lighting in unstructured-mesh rendering for three data sets. Left: Lighting helps discover turbulent flow in the vicinity of the wing that cannot be observed otherwise. Proper gradient estimation should highlight these shapes without adding extraneous artifacts. Right: Lighting also helps understand the spatial relationship between different occluding isosurfaces. With lighting, the actual shape of the semi-transparent isosurfaces can be easily perceived even in the presence of overlapping surfaces.

Hybrid gradient reconstruction improves the visual quality by applying different methods in a single mesh according to a given quality metric. In our experiments, regression-based methods behave better for irregular elements in comparison to methods based on averaging, and vice versa for regular elements. This suggests that the choice of method should follow the local mesh quality instead of being applied globally. Fixed-size neighborhood gradient reconstruction is another heuristic that ranks the neighbor vertices of a given point so that the gradient can be reconstructed as accurately as possible without incurring in much overhead.

2 RELATED WORK

Volume rendering of unstructured meshes has become an important tool for understanding computational fluid dynamics and mesh discretizations of PDEs. The most predominant rendering approaches are *cell projection* [27], *point-based approaches* [32], [34] and *raycasting* [11], [30]. For its simplicity, some practitioners *re-sample* the unstructured mesh into a regular grid and render this grid directly [29], [31]. GPU-based implementations of these methods exist [9], [18], [30]. Cell projection and point-based approaches, often classified as object-order approaches, do not require an explicit connectivity of the cells. However, they require visibility sorting. Image-order approaches, such as raycasting, do not require visibility sorting, but they require the connectivity information to traverse the cells along the view rays. In this paper, we use raycasting to test our results and provide a visual comparison. Our implementation is based on the implementations by Garrity [11] and Weiler et al. [30]. The methods described in this paper and the results of our evaluation, however, are applicable to both object- and image-order approaches.

The study of gradients in unstructured meshes can be understood from both the simulation and the visualization standpoints. In simulation, the study of gradient reconstruction methods leads to more accurate reconstructions of an underlying scalar function and better error bounds for the discretization of PDEs. Most of these studies rely on methods based on linear regression and the Green-Gauss theorem [7], [19], [17]. Aftosmis et al. discuss the behavior of linear reconstruction methods on unstructured meshes [3]. In their studies, Barth [7], Mavriplis [17] and Anderson [5] found that inverse distance weighting has a significant impact on linear regression models for gradient estimation, while methods such as Green-Gauss degrade. In an attempt to improve the error bounds of gradient reconstruction, Shewchuk studies the impact of cell shape in linear reconstruction and provides a series of quality metrics for tetrahedral cells [26]. Petrovskaya [25] and Apel et al. [6] also study the impact of cell shape in reconstruction algorithms. While there is no consensus on what is a good mesh element, these studies suggest that these methods produce noticeable differences in the gradient reconstruction as the mesh becomes more irregular. In this paper, we aim at validating some of these findings from the rendering standpoint.

In visualization, gradient estimation becomes important as shading is an essential cue for shape. For geometric objects, such as triangle meshes, normals to the surface can be computed directly from the geometric representation. Volume representations, however, do not encode explicit geometry but are rather sampled in a grid or an unstructured mesh. The gradient is not computed directly from the geometric information, but requires the consideration of the volume data. Yagel et al. surveys these methods for structured grids and classifies them into image-space and object-space methods [33]. With the advent of fast graphics processors, object-space methods became the norm. Möller et al. compare normal estimation schemes from the point of view of the quality of the reconstruction filter [20]. A similar study is carried out by Bentum et al. in the frequency domain [8]. In the spirit of generalization, Thürmer and Wüthrich consider the normal computation in 3D space as an approximation resulting from sampling a spatial neighborhood of each point, and describe the importance of variable weighting of each sample [28]. Neumann et al. also consider the problem around a neighborhood and pose the problem as 4D regression [22]. Although these two methods were described for structured grids, their derivations also apply to unstructured meshes. While not designed for volume rendering, the need for reconstructing smooth functions from unorganized points has emerged in the point-based rendering community. Estimating the gradient to a surface has been posed as a total least squares problem [12] or a moving least squares problem [24]. Similar techniques can be applied when we consider an unstructured mesh as a collection of points.

The need for cell-based gradient estimation surfaced in the visualization of unstructured meshes to speed-up the sampling of the scalar fields at arbitrary points within a cell. This was first proposed by Garrity [11], who estimated the constant cell gradient via a linear approximation. This was later used by Weiler et al. in GPU-based raycasting [30]. Constant cell gradient, however, is not adequate for lighting, and node-centered gradients are needed. Cignoni et al. use the average of the gradient of the incident cells to compute the gradient at a node [10], and use it to render shaded isosurfaces. Ma et al.

compute the gradient based on an approximation of the Green-Gauss theorem [15], which was later modified by Meredith and Ma, who use the directional derivatives of the scalar field to obtain a fast approximation [18]. Levy et al. use unweighted linear regression to estimate the normals [14]. We show that weighted regression provides a better estimate of the gradient than unweighted schemes. In recent approaches, the lack of lighting is compensated with opacity transformations, which result in an appearance similar to shaded isosurfaces [21].

This heterogeneity of methods in both the simulation and the visualization communities demonstrates that there is no consensus on what are the most adequate methods for adding lighting to volume data in unstructured meshes. In this paper, we provide a quantitative and qualitative evaluation of the most prominent linear gradient reconstruction methods. We seek to guide future generations of unstructured mesh visualization systems towards high-quality volume rendering.

3 LINEAR GRADIENT RECONSTRUCTION

Let us define an unstructured mesh as a collection of connected points $\mathbf{x}$ that discretize a scalar field f. The linear approximation of this function at a given point $\mathbf{x}_0 + \mathbf{h}$ is given by

$$f(\mathbf{x}_0 + \mathbf{h}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot \mathbf{h} + O(||\mathbf{h}||^2) \quad (1)$$

where $\nabla(f(\mathbf{x}_0)$ is the gradient at point $\mathbf{x}_0$ and $\mathbf{h}$ is a discretization step. The goal of gradient estimation is therefore to recover the function ∇f such that Eq. (1) holds for any given point. Because the approximation is linear, these methods are collectively known as linear gradient reconstruction methods. We can further classify these methods into two groups: averaging-based methods, which construct the gradient as a weighted average of the neighboring gradients, and regression-based methods, which posit Eq. (1) as a least squares problem.

To understand the sources of error in these methods, we can expand the second term of the linear approximation. The residual of $r_2 = f - \hat{f}$, where $\hat{f}$ is the linear approximation of f, is

$$r_2(\mathbf{x}_0 + \mathbf{h}) = \frac{1}{2!}\mathbf{h}^\top \nabla^2 f(\mathbf{x}_0)\mathbf{h} + O(||\mathbf{h}||^3) \quad (2)$$

where $\nabla^2 f$ is the Hessian matrix of f. Furthermore, the absolute error can be bounded as:

$$||r_2|| \le ||\mathbf{h}||^2 ||\nabla^2(f(\xi, \zeta, \eta))|| \quad (3)$$

for some $(\xi, \zeta, \eta) \in (\mathbf{x}_0, \mathbf{x}_0 + \mathbf{h})$. Therefore, linear approximation methods are both dependent on the mesh discretization and the complexity of the scalar field. This is true for both structured and unstructured meshes. Unlike structured grids, unstructured meshes have a variable discretization distance $\mathbf{h}$. Therefore, the shape of the mesh element is also a factor. Numerous quality metrics have been proposed for tetrahedra, as described in [26]. The study of these metrics, of which the most common are aspect ratio and the ratio of the inscribing sphere and maximum edge length, has led to tighter bounds on the approximation error [6]. Here, we are not concerned about these bounds, but rather in the effects of the different factors in the volume rendered image. Mavriplis showed that regression-based methods provide better estimates on irregular elements rather than averaging-based methods [17]. Shewchuk also notes that this effect has been misunderstood as due to elements of poor aspect ratio, but argues that it is the presence of large angles that results in larger approximation error. In our visual analysis, we show how the different methods behave differently depending on the element shape.

3.1 Averaging Based Methods

In this family of methods, the gradient is computed as a weighted average of functions of the gradient or scalar values at a neighborhood around a node. In general, this can be expressed as the linear combination

$$\nabla f(\mathbf{x}_0) = \sum_i w_i \nabla f(i) \quad (4)$$

where w_i is a weighting factor, and $\nabla f(i)$ is the constant gradient at a cell i. The gradient at a cell can be computed by considering Eq. (1) for the 4 vertices of a tetrahedron, here denoted as column vectors $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, resulting in the 3×3 linear system

$$\begin{bmatrix} (\mathbf{x}_1 - \mathbf{x}_0)^\top \\ (\mathbf{x}_2 - \mathbf{x}_0)^\top \\ (\mathbf{x}_3 - \mathbf{x}_0)^\top \end{bmatrix} \nabla f = \begin{bmatrix} f(\mathbf{x}_1) - f(\mathbf{x}_0) \\ f(\mathbf{x}_2) - f(\mathbf{x}_0) \\ f(\mathbf{x}_3) - f(\mathbf{x}_0) \end{bmatrix} \quad (5)$$

The left hand side consists of a 3×3 matrix where each row is a displacement and the three columns are the components in each of the spatial dimensions. The right hand side is a column vector of scalar differentials. This system can be solved exactly for non-degenerate tetrahedra, i.e., tetrahedra that do not collapse into a plane, a line or a point.

3.1.1 Cell Weighting

Since the cells around a given vertex are not of the same shape, the weighting factors w_i can be computed to give higher importance to those cells that should contribute more to the average gradient (Fig. 2(a)). Here, we consider four methods:

Uniform. This is the case when all cells are weighted uniformly. This method is the most commonly used in volume rendering due to its simplicity, but does not adapt to meshes of varying shape.

Volume. Each cell is weighted according to its own volume. Although it adapts better to meshes of varying shape, some elements may exhibit small aspect ratio while having the same volume of other more regular elements. Later on, we show that this method is equivalent to obtaining the gradient using the Green Gauss theorem.

Solid Angle. A cell is weighted by the solid angle subtended by the cell at the central vertex $\mathbf{x}_0$, measured as the surface area of a unit sphere covered by the opposite face to vertex $\mathbf{x}_0$.

Inverse centroid distance. Each cell is weighted by the inverse of the distance between the central vertex $\mathbf{x}_0$ and the centroid of cell i.

3.1.2 Green-Gauss Method

A different derivation of the gradient is obtained using the Green-Gauss theorem, which states that for a volume Ω enclosed by a surface S,

$$\int_\Omega \nabla f d\Omega = \int_{\partial\Omega} f\mathbf{n}dS \quad (6)$$

where $\mathbf{n}$ denotes the outward pointing normal vector to the surface S, as shown in Fig. 2(b). In an unstructured mesh, the average gradient at a node can be approximated by

$$\nabla f(\mathbf{x}_0) \quad \approx \quad \frac{1}{|\Omega|}\int_{\partial\Omega} f\mathbf{n}dS \quad (7)$$

$$\approx \quad \frac{1}{|\Omega|}\sum_{i \in S_1, \ldots, S_n} \overline{f}_i \mathbf{n}_i \quad (8)$$

The first approximation replaces the volume integral of the region enclosing the vertex by the total volume. The second
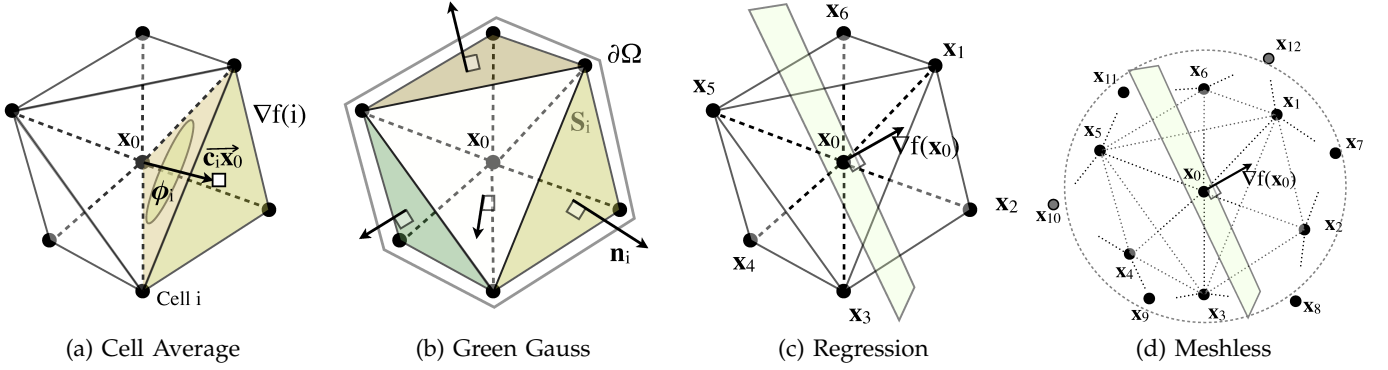
Fig. 2. Overview of the gradient estimation methods. (a) Weighted average of the neighboring cell gradients. Typical weights are volume, solid angle ϕ_i or inverse centroid distance $1/||\mathbf{c}_i - \mathbf{x}_0||$. (b) The Green Gauss method approximates the gradient as the surface integral of the control volume Ω. (c) Regression fits a plane for the gradient based on the contribution of the direct neighbors of vertex $\mathbf{x}_0$. (d) Meshless methods apply regression on scattered points in a spatial neighborhood (e.g., a sphere of radius r).

approximation is done over the surface integral using the trapezoidal rule on each of the faces S_i defining the surface. $\mathbf{n}_i$ denotes the outward pointing normal vector of the face S_i. The scalar value at a face, $\overline{f}_i$, is obtained as the linearly interpolated scalar value at the barycenter of face S_i.

We can see that the Green Gauss method is equivalent to computing the volume weighted cell average gradient. Let us define the volume integral of the gradient in a neighborhood of cells around a central vertex $\mathbf{x}_0$. Assuming that the gradient at the cell is constant [3].

$$\int_\Omega \nabla f d\Omega = \sum_i \nabla f(i) \int_{\Omega_i} d\Omega \quad (9)$$

$$= \sum_i \nabla f(i) V_i \quad (10)$$

where V_i is the volume of cell i. That is, if we convert each element volume integral into an element Green-Gauss surface integral, the contributions from shared internal faces will cancel out in the summation over the entire region, resulting in the Green-Gauss approximation.

3.2 Regression Based Methods

Another family of methods can be derived from Eq. (1) by fitting a hyperplane that best satisfies the equation for a number of sample points, as depicted in Fig. 2(c). In the case of node-centered gradients, Eq. (1) can be generalized to an over-constrained system of equations.

$$\begin{bmatrix} (\mathbf{x}_1 - \mathbf{x}_0)^\top \\ (\mathbf{x}_2 - \mathbf{x}_0)^\top \\ \vdots \\ (\mathbf{x}_k - \mathbf{x}_0)^\top \end{bmatrix} \nabla f = \begin{bmatrix} f(\mathbf{x}_1) - f(\mathbf{x}_0) \\ f(\mathbf{x}_2) - f(\mathbf{x}_0) \\ \vdots \\ f(\mathbf{x}_k) - f(\mathbf{x}_0) \end{bmatrix} \quad (11)$$

$$(12)$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are the vertex neighbors of vertex $\mathbf{x}_0$. Equivalently, this system can be expressed in matrix form,

$$\mathbf{X}\nabla f = \mathbf{b} \quad (13)$$

where $\mathbf{X}$ is a $k \times 3$ matrix whose columns are the displacement of each vertex in the spatial dimensions, and $\mathbf{b}$ is a column vector of dimensions $k \times 1$ of scalar value differentials. The problem can be solved using linear least squares.

As can be seen, this method extends naturally to arbitrary element shapes and neighborhoods. In particular, considering

all the vertices in a spatial neighborhood of $\mathbf{x}_0$ leads to a meshless gradient reconstruction scheme. The same cannot be said about averaging methods. As pointed out by Mavriplis, the Green Gauss approximation is generally not exact for discretizations other than tetrahedra [17], although different control volumes can be defined for such cases.

3.2.1 Weighting

To account for the unstructured nature of the mesh, regression-methods can be modified to add weights to each of the vertex neighbors. Let w_i denote a weighting factor associated with vertex $\mathbf{x}_i$. The gradient reconstruction can be posed as the over constrained system:

$$\mathbf{W}\mathbf{X}\nabla f = \mathbf{W}\mathbf{b} \quad (14)$$

where $\mathbf{W} = diag\{w_i\}$ is a $k \times k$ diagonal matrix containing the weights of all k neighbors of vertex $\mathbf{x}_o$. The solution to this system can be computed using weighted least squares, solving the 3×3 system

$$\mathbf{X}^\top \mathbf{W}^2 \mathbf{X}\nabla f = \mathbf{X}^\top \mathbf{W}^2 \mathbf{b} \quad (15)$$

With regards to weighting, we consider two cases: *unweighted regression*, for $w_i = 1$, commonly used in volume rendering, and *inverse distance weighted* regression, in which case $w_i = \frac{1}{||\mathbf{x}-\mathbf{x}_0||^2}$, where $|| \cdot ||$ denotes the Euclidean norm of a 3D vector. Mavriplis showed that, in general, weighted regression provides better estimates than unweighted regression, especially for irregular elements [17]. We can see this in the norm of the matrix in Eq. (15). The matrix is inherently dependent on the element shape. For irregular elements, the difference in edge lengths generates a large conditioning for the matrix. For unweighted regression, the determinant of $\mathbf{X}^\top \mathbf{X}$ grows as $O(||\mathbf{h}||^6)$ (since it grows as the cube of the elements in the matrix, which are $O(||\mathbf{h}||^2)$) and the problem is ill-conditioned for nearly coplanar cells. Weighting using inverse distance cancels out the dependency on the shape of surrounding tetrahedra, and the determinant of $\mathbf{X}^\top \mathbf{W}^2 \mathbf{X}$ grows as $O(1)$, so that it is less sensitive to near-coplanar cases. This difference in accuracy was observed in our experiments and is consistent with previous results [17], [25].

3.2.2 4D Regression

As an alternative to 3D regression, Eq. (1) can be formulated as a 4D regression problem, as suggested by Neumann et al. [22].

$\delta = 0.0$ $\delta = 0.125$ $\delta = 0.25$ $\delta = 0.5$
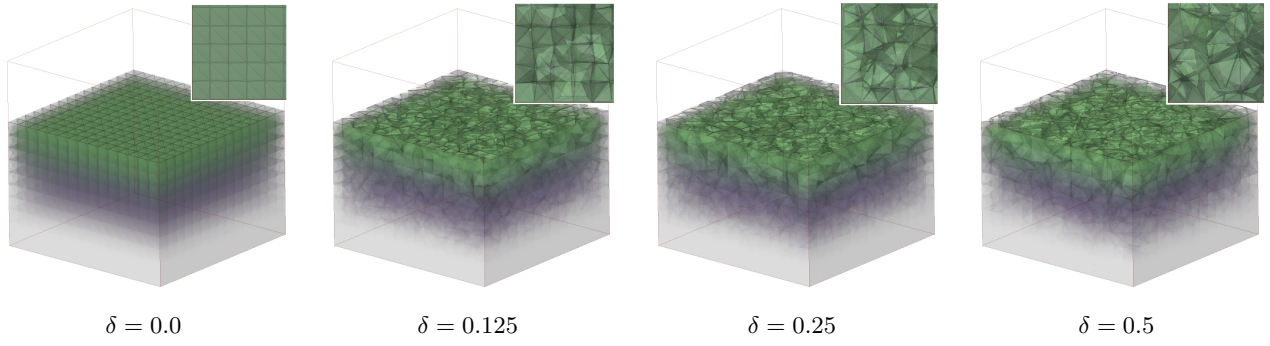
Fig. 3. Example synthetic data sets for $N = 16$ and increasing irregularity factors. We render the volume with constant interpolation through the cells to highlight the shape of the mesh elements

In this case, the scalar value at a given vertex is considered as an unknown along with the gradient, and results in the 4×4 system of equations:

$$
\begin{bmatrix}
(\mathbf{x}_1 - \mathbf{x}_0)^\top & 1 \\
(\mathbf{x}_2 - \mathbf{x}_0)^\top & 1 \\
\vdots & \\
(\mathbf{x}_k - \mathbf{x}_0)^\top & 1
\end{bmatrix}
\begin{bmatrix}
\nabla f \\
f(\mathbf{x}_0)
\end{bmatrix}
=
\begin{bmatrix}
f(\mathbf{x}_1) \\
f(\mathbf{x}_2) \\
\vdots \\
f(\mathbf{x}_k)
\end{bmatrix}
$$

The solution contains the gradient and also a *filtered* value of the scalar at that point. Unlike 3D regression, weighting based on inverse distance is not as effective, since the last column of ones gets replaced by the inverse distance, which may be large for irregular elements.

3.2.3 Meshless Regression

In general, if we consider all points in the neighborhood of a given vertex, the regression method does not need the explicit mesh connectivity anymore. This observation has led to meshless reconstruction methods, such as the raycasting method presented by Ledergerber et al [13]. According to this method, a scalar function f can be approximated as the linear combination of a set of basis functions:

$$
f(\mathbf{x}) = \mathbf{g}(\mathbf{x})^\top \mathbf{c_x} \tag{16}
$$

where $\mathbf{g}$ is a set of basis functions and $\mathbf{c_x}$ is a set of coefficients, found using regression. For linear approximations, i.e., $\mathbf{g}(\mathbf{x}) = [x, y, z, 1]^\top$ or $\mathbf{g}(\mathbf{x}) = [x, y, z]^\top$, the result is equivalent to 4D and 3D regression, as described above. The coefficient is then

$$
\mathbf{c_x} = \left(\mathbf{X}^\top \mathbf{W}^2 \mathbf{X} \right)^{-1} \mathbf{X}^\top \mathbf{W}^2 \mathbf{b} \tag{17}
$$

The gradient can then be obtained as the partial derivatives of this function with respect to the spatial coordinates, as pointed out Ledergerber et al [13].

$$
\begin{aligned}
\frac{\partial f(\mathbf{x})}{\partial x_k} &= \frac{\partial \mathbf{g}(\mathbf{x})}{\partial x_k}^\top \mathbf{c_x} + \mathbf{g}(\mathbf{x})^\top \frac{\partial \mathbf{c_x}}{\partial x_k} \tag{18} \\
&= \frac{\partial \mathbf{g}(\mathbf{x})}{\partial x_k}^\top \mathbf{c_x} \\
&\quad - \mathbf{g}(\mathbf{x})^\top \mathbf{A}(\mathbf{x})^{-1} \left(\frac{\partial \mathbf{A}(\mathbf{x})}{\partial x_k} \mathbf{c_x} - \mathbf{X}^\top \frac{\partial \mathbf{W}^2(\mathbf{x})}{\partial x_k} \mathbf{b} \right)
\end{aligned}
$$

where $\mathbf{A}(\mathbf{x}) = \mathbf{X}^\top \mathbf{W}^2(\mathbf{x}) \mathbf{X}$. Lederberger et al. point out that the first term of this derivative is a good approximation of

the gradient. For the case of linear basis functions, the resulting derivatives are the coefficients corresponding to the basis functions x, y and z. Therefore, approximating the gradient using the first term of the derivative of f is equivalent to the regression methods shown in the previous sections. In the evaluation sections, we compare meshless methods with those based on an explicit mesh while varying the size of the neighborhood.

4 EVALUATION

As shown in the previous section, the accuracy of the gradient reconstruction filter depends on both the resolution of the mesh and the complexity of the scalar field. In addition to resolution, the shape of the mesh also contributes to the accuracy. We ran a series of experiments to measure the effect of the choice of a particular gradient method on the accuracy of reconstruction, according to a given variable. The variables we consider are: (1) mesh resolution, defined as the discretization distance of a regular grid, (2) irregularity factor, defined as the maximum deviation of the vertices from the regular grid, (3) element shape, defined as the maximum aspect ratio of the elements in the mesh and (4) complexity of the scalar field, measured as the maximum magnitude of the Hessian of the scalar field.

4.1 Experimental Data Sets

Our experiments consist of both synthetic and "real" meshes. The synthetic mesh benchmark consists of a series of meshes obtained from a regular grid. We control two variables: the size of the grid, N, which indicates the discretization distance $||\mathbf{h}|| = 1/N$, and the irregularity factor δ, which indicates a maximum random deviation of the vertices in the mesh from the grid points. The grid is then defined as a collection of points (connected using tetrahedral elements) defined as:

$$
\mathbf{x}_{ijk} = (i + r_x(\delta), j + r_y(\delta), k + r_z(\delta))\mathbf{h} \tag{19}
$$

for $i, j, k \in \{1, \dots, N\}$. Functions $r_x(a), r_y(a), r_z(a)$ are random number generators in the interval $[0, a]$. For $\delta = 0$, the mesh is a regular grid. Fig. 3 shows four example meshes for $N = 16$ and different values of δ. In our experiments, we tested 40 of these meshes, for sizes $N = \{8, 16, \dots, 64\}$ and deviation factors $\delta \in \{0.0, 0.125, 0.25, 0.375, 0.5\}$. Although δ makes the elements deviate from the grid, they are, in average, close to regular. To model highly irregular elements, we created a subset of these meshes where the z dimension is scaled down by a decreasing factor. This generates a series of elements of small aspect ratio and large angles as the element
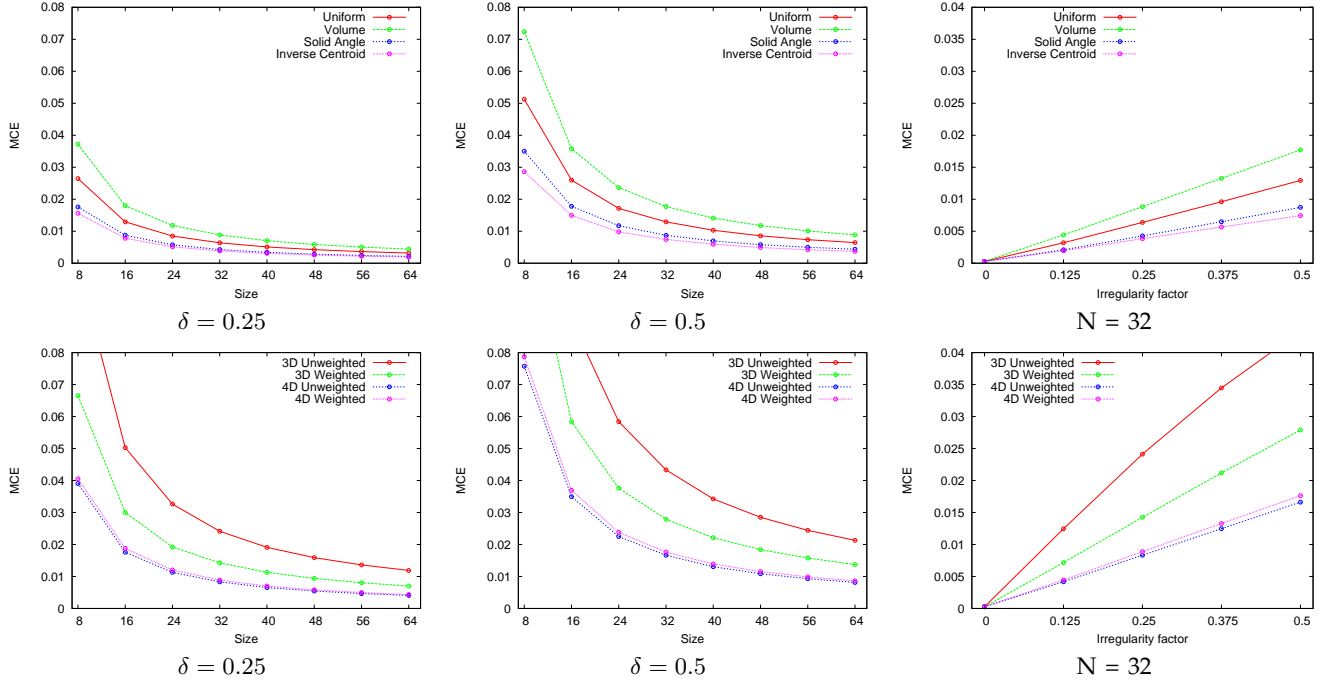
Fig. 4. Gradient reconstruction accuracy (MCE) in terms of mesh resolution and regularity for a spherical scalar field. Top: Average weighting methods, for two meshes of decreasing regularity. Error decreases quadratically with mesh resolution, and linearly with irregularity (right). For a regular mesh ($\delta = 0$), the error is very small, but non-zero. Bottom: Regression-based methods. Notice the disparity between 3D weighted and unweighted regression. Weighting has little effect for 4D regression.

shape approaches a plane. For scalar fields, we used two types of analytical functions. A spherical function in a unit cube (with bounding box from $(0,0,0)^\top$ to $(1,1,1)^\top$, defined as $f(\mathbf{x}) = ||\mathbf{x} - (0.5, 0.5, 0.5)^\top||$, and the Marschner-Lobb function, as defined in [16]. These scalar functions allows us to evaluate the accuracy of the gradient reconstruction methods as we can find the ground truth gradients analytically. To validate our results in "real" datasets, we compiled a series of meshes from flow simulation and tetrahedralizations. Both types of meshes contain a mix of low and high quality elements. For CFD simulations, the use of elements of varying shape allows the mesh to align to the flow. For tetrahedralized models, irregular elements are required to adapt the mesh to the shape of the enclosing surface. Table 1 summarizes the datasets compiled for our experiments and their corresponding statistics.

4.2 Quality Metrics

To measure the quantitative accuracy of each method, we use the mean cosine error (MCE), defined as follows:

$$MCE \quad = \quad \frac{1}{N} \sum_{i=1}^{N} \cos^{-1} \mathbf{n}_i \cdot \hat{\mathbf{n}}_i$$

where $\mathbf{n}_i$ and $\hat{\mathbf{n}}_i$ are the exact and estimated normals at a given point, respectively. One of the problems with this metric is the inability to represent the variance of the samples. For this reason, we derived a correlation metric, based on the contribution of the gradient to lighting. In this case, we define a random variable as the diffuse component of a point with a directional light at $\mathbf{l} = (1,1,1)$, and then used the Pearson coefficient as the quality metric between the ground-truth diffuse component, $\mathbf{n}_i \cdot \mathbf{l}$, and the approximation resulting from a gradient approximation, $\hat{\mathbf{n}}_i \cdot \mathbf{l}$. This helps us detect different degrees of variability among the four methods.
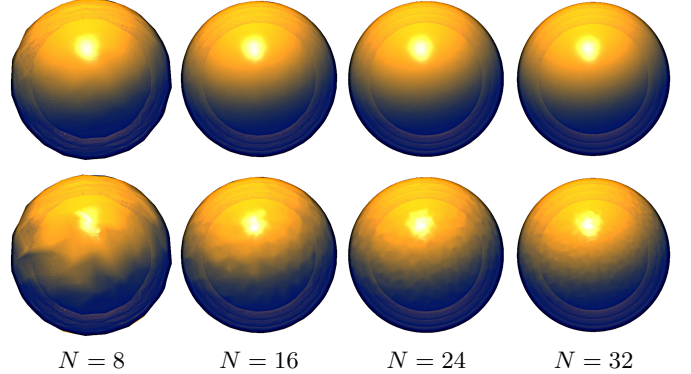

Fig. 5. Visual comparison of inverse centroid weighted average (top) and unweighted 3D regression (bottom) for meshes of increasing resolution and irregularity factor $\delta = 0.25$. Unweighted regression results in a bumpy appearance and the mesh elements are evident when compared to averaging.

4.3 Effects of Mesh Resolution

In the first experiment, we generated a series of synthetic meshes with varying resolution, as described above.

Fig. 4 shows the gradient reconstruction error for a sphere scalar field in relation to the resolution of the mesh. As N increases, the discretization distance decreases, improving the accuracy of the reconstruction. Notice that solid angle and inverse weighted centroid distance produce better estimates than uniform weighting. Volume weighting, i.e. Green-Gauss reconstruction provides the least accurate reconstruction. Notice also the quadratic trend as the discretization distance decreases. When we keep the discretization distance constant and vary the irregularity factor, we notice a linear trend.

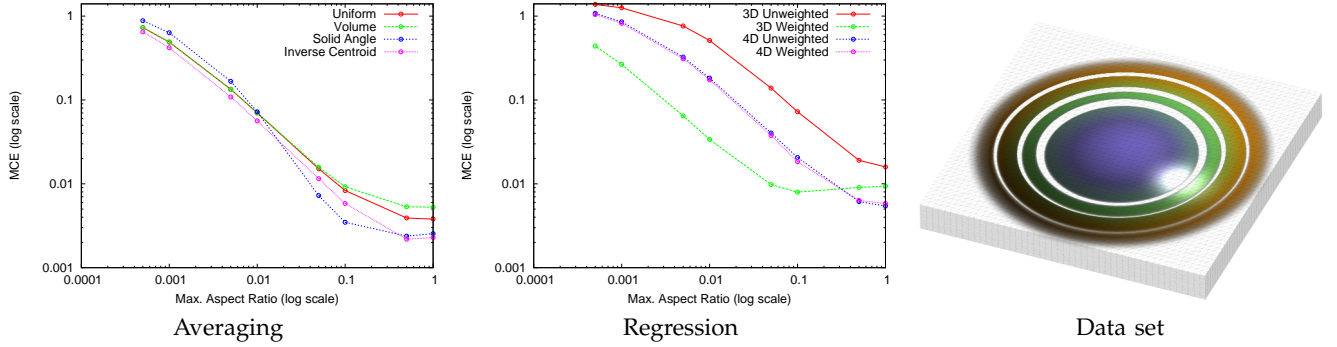| Averaging | Regression | Data set |

Fig. 6. Effects of element shape on accuracy. For averaging methods, solid angle appears to be more accurate, but it becomes less effective as the mesh becomes highly irregular. Inverse centroid remains as the most accurate. For the case of regression methods, 3D weighted regression outperforms the other methods and becomes even more accurate than averaging-based methods.
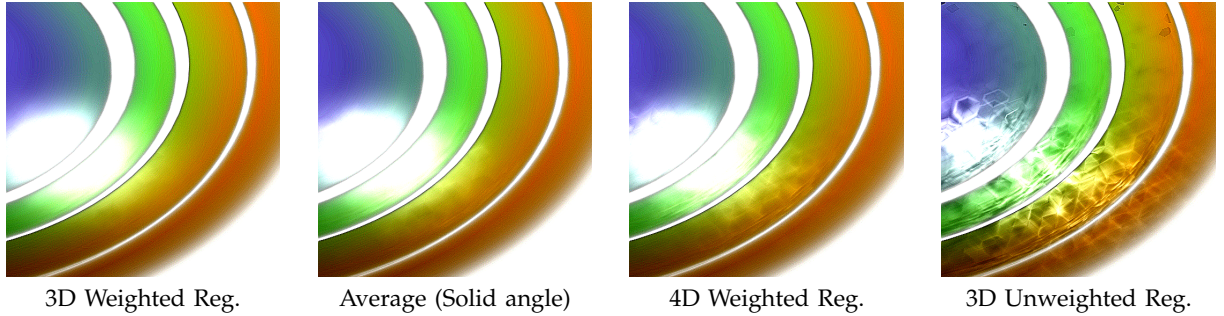

| 3D Weighted Reg. | Average (Solid angle) | 4D Weighted Reg. | 3D Unweighted Reg. |

Fig. 7. Visual comparison of the different methods with respect to element shape (Here, the max. aspect ratio is 0.01). Notice the bumpiness of other methods compared to 3D weighted regression as suggested by the specular highlights. 3D unweighted regression leads to poor estimates that impacts negatively the benefit of the rendering.

Fig. 5 shows the progression of gradient accuracy for the purposes of lighting as the resolution of the mesh increases. On top, we show the effects of inverse centroid weighted average. At the bottom, we show the results for unweighted 3D regression. Though having the same geometry, there is a discernible bumpiness in the appearance of the spheres for the case of regression, and the meshing quality is apparent. For weighted average, the sphere looks smooth.

4.4 Effects of Element Shape

As noted above, although the mesh is irregular, the average element has a regular shape. To measure the effects of element shape, we created a series of meshes where we scale down one of the dimensions of the mesh, thereby decreasing the aspect ratio of the elements. Fig. 6 shows the mean cosine error for the sphere scalar function for both averaging and regression-based methods. Note the particular behavior of solid angle weighting vs. volume weighting (or Green Gauss). For better shaped meshes, it appears more accurate than Green Gauss, but this trend changes as the shape quality decreases. Overall, however, the accuracy of averaging methods seems to converge.

For regression-based methods the difference is more dramatic. We can clearly see the effects of weighting, as reported by Mavriplis [17]. For low quality meshes (in this case for stretch factor below 0.05), weighted 3D regression performs even better than averaging methods. Unweighted 3D regression performs consistently poorly in comparison. For the case of 4D regression, weighting has no apparent effect.

To visualize these results, we show the rendering results for a spherical data set in a highly irregular mesh (Fig. 7). Weighted

regression, as suggested by the quantitative results, provides the higher quality. Notice the smoothness of the color on the surface and in the specular highlights. It is followed closely by averaging methods. The most noticeable difference of these is the solid angle averaging. Notice a bumpier appearance in the specular reflections. 4D regression provides a poorer estimate of the gradient, but the worst quality are obtained with unweighted 3D regression.

4.5 Effects of Neighborhood Size

As pointed out above, regression-based models can be easily extended to estimate the gradient based on an arbitrary neighborhood, instead of the neighbors given by the mesh connectivity. This meshless approximation requires a spatial neighborhood around each vertex. Since the mesh density may change dramatically from vertex to vertex, this spatial neighborhood is usually defined adaptively. To compare this method with mesh-based reconstruction, we define the neighborhood as a variable isotropic radius R_C proportional to the radius of the sphere circumscribing the immediate neighbors of a vertex (as defined by the mesh), as depicted in Fig. 2(d). Therefore, when the support radius $R = 1.0R_C$, the meshless approximation contains all the points that are considered in a mesh-based approximation (although it may contain more), and we expect to see similar accuracy. In general, anisotropic weights are more useful to account for the variation in point density in the mesh. The evaluation of this aspect and the accuracy of meshless gradient reconstruction is well beyond the scope of this paper. Here, we consider the gradient reconstruction using linear basis functions and isotropic neighborhoods. As the radius

increases, the accuracy of the linear meshless approximation improves, as more points are considered. However, at some point the accuracy is expected to decrease, since it approximates a larger region with a plane. Fig. 8 shows the result of the gradient of the local meshless approximation and the analytical derivatives, i.e. the first term and the full expression in Eq. (18), respectively. The plot also shows the result of averaging (using inverse centroid weighting) and weighted regression, computed once using the connected neighbors of a vertex, as baselines. On the top, we see the accuracy for a spherical function, as describe above, and the bottom shows the accuracy for the Marschner-Lobb data set. We can see that meshless reconstruction becomes as accurate (or better) than average-based methods for certain neighborhood sizes. The accuracy is much better for a smooth function, such as a sphere, compared to the Marschner-Lobb function, which exhibits a higher spatial frequency. We see that as the number of neighbors increases, the accuracy of the reconstruction starts to decrease. This is more noticeable for the latter dataset, since the reconstruction "flattens" the surface (it approximates the region with a plane) and deviates considerably from the ground truth gradient.

Although the numerical result is not as accurate, the visual effect of adding more points is a smoother surface. Fig. 9 shows three volume rendered images for increasing neighborhood radii. The increase of neighborhood results in smoother images, while retaining the overall shape of the isosurfaces. Note in particular the disappearance of artifacts near the contours of the sinusoidal ridges. We compare the reconstruction using the regression approximation (top) with the one obtained as the analytical gradient of the meshless reconstruction (bottom). Consistent withe results in Ledergerber et al. [13], the difference is only noticeable for models presenting high frequencies. Note that the meshless reconstruction was only performed on the gradient and not the function itself, for the purposes of evaluation. In general, meshless raycasting approximates both the function and its gradient.

4.6 Effects of Scalar Field Complexity

To test the effects of scalar field complexity, we kept the size and shape of the mesh fixed and change the complexity of the scalar function. To this purpose, we used a synthetic grid of $N = 48$ and $\delta = 0$ and $\delta = 0.25$. For the scalar field, we used a Marschner-Lobb function, as described in [16], while varying α and f_M. The trace of the Hessian $Tr(H(f))$ grows proportional to the product of these two parameters. Fig. 10 shows the mean cosine error for several Marschner-Lobb functions, labeled ij, where $\alpha = 0.1i$ and $f_M = 0.1j$. Notice that, in overall, the accuracy decreases as the complexity of the scalar function (measured as the trace of the Hessian) increases. There are some cases, however, where the accuracy improves. This is noticeable in the regular grid more than the irregular mesh. This can be explained as a case of alignment of the scalar field with the mesh, as noted by Shewchuk [26]. When we plot correlation, however, we see an overall increasing trend, accounting for the variability in the gradient reconstruction. Also, we noted that 3D weighted regression appears more correlated than 4D unweighted regression, even though the mean error appears larger. Again, we get a systematic error that, although deteriorates the average error, appears correlated.

4.7 Quantitative Comparison

To test in "real" meshes, we applied the reconstruction to a series of meshes from CFD simulations and finite element
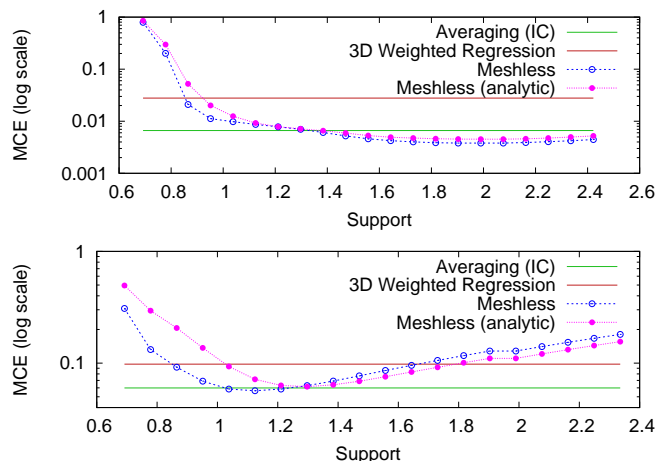

Fig. 8. Effects of neighborhood size for a mesh with a spherical scalar function (top) and the Marschner-Lobb function (bottom). We plot the error of meshless methods (approximation and analytical derivatives) as the support radius increases, and compare it to regression and averaging methods. For a support radius factor near 1, the error is similar to that of mesh-based regression. Larger support improves the reconstruction momentarily, until the real shape of the scalar function deviates from the linear approximation (a plane).

Mesh	# Vertices	# Cells	Source
spx	2896	12936	Nitrosso, Electricité de France
fighter	13832	70125	Neely and Batina, NASA.
hand	28796	125127	Meshing, Alliez et al. [4]
skull	37813	156135	Meshing, Alliez et al. [4]
parikh	103064	567862	Airflow sim.,ViGYAN,Inc.
post	108300	616050	Liquid Oxygen Post.
heart	140425	689020	Tetrahedral model of heart [2]
bucky	262144	1250235	Simulation, Haimes, MIT
plasma	274625	1310720	Simulation, ISTI [1]
m6	357900	2000034	ONERA M6 Wing, NASA
sf2	378747	2067739	Earthquake Simulation [23]
lwt	804056	4607888	LWT aircraft airflow simulation

TABLE 1
Summary of data sets for our quantitative and visual analysis.

simulations. Fig. 11 shows the result for both types of meshes on a sphere scalar function. By looking at this result, we can confirm the trends exhibited in our previous experiment. In general, averaging methods perform better than regression. Furthermore, inverse centroid distance appears more accurate than other weighting methods, and volume weighting (Green Gauss) provides the least accurate reconstruction. A notable exception is the *heart* data set. Further inspection showed the presence of near degenerate tetrahedra due to meshing errors. Interestingly, weighted 3D regression also performs better when compared to other regression methods for this data set, as with the *skull* and *hand* data sets. These three data sets are tetrahedralizations of 3D surfaces. For CFD simulations, a notable case is the *post* data set, where 3D weighted regression performs better than other regression methods. This data set is characterized by the varying element shapes and low quality of the mesh near the outer boundary regions.
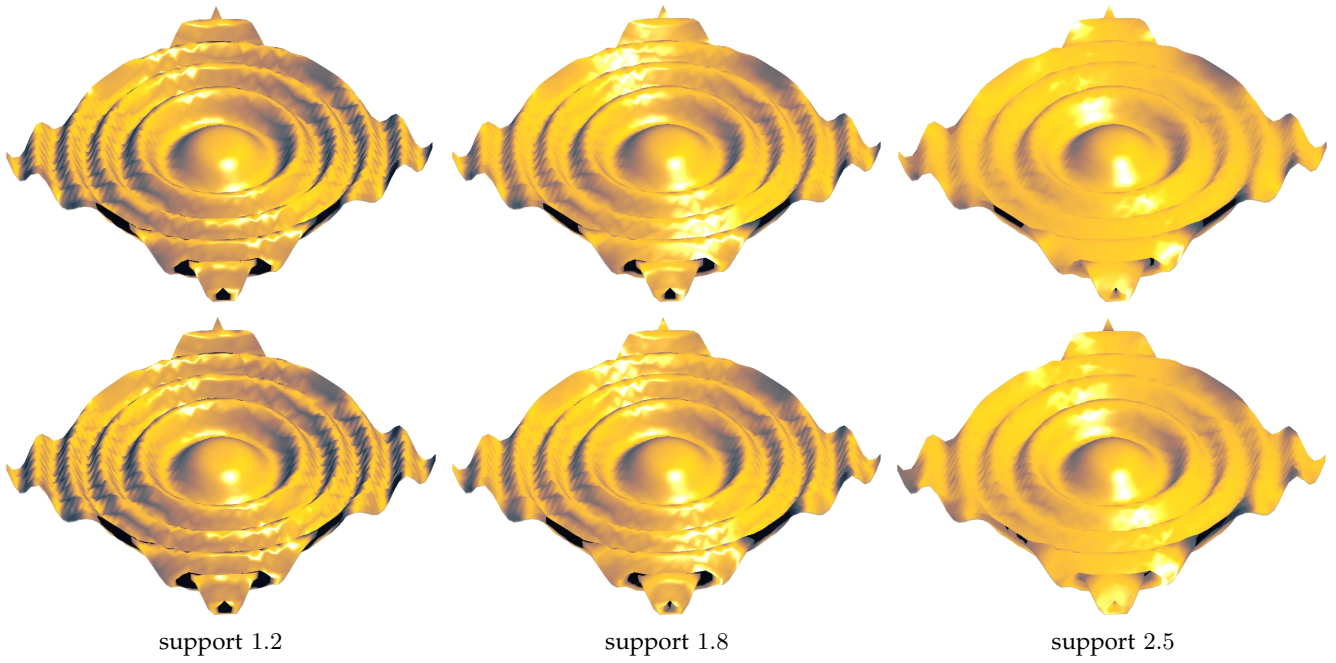
support 1.2   support 1.8   support 2.5

Fig. 9. Visual comparison on the effects of neighborhood size for the Marschner-Lobb scalar function. As we introduce more points, the rendered surfaces become smoother and artifacts due to meshing become less apparent. Increasing the neighborhood further makes the surface to appear flatter. Top: Local meshless approximation. Bottom: Analytical derivatives of meshless reconstruction.
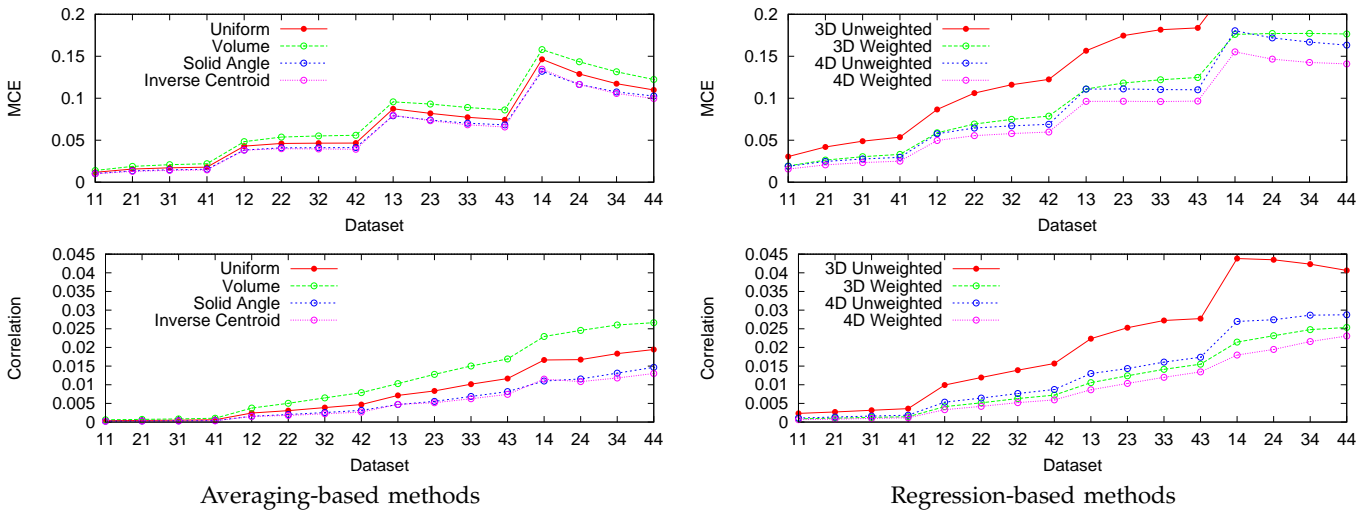

Averaging-based methods   Regression-based methods

Fig. 10. Effects of scalar field complexity. Averaging-based methods are in general more accurate. Of the regression-based methods, 4D weighted is the only comparable, which estimates similar to volume weighting. When these methods are compared in terms of correlation, we notice that 3D weighted regression has a better estimate (correlated error) than 4D unweighted regression. 3D unweighted regression is significantly less accurate.

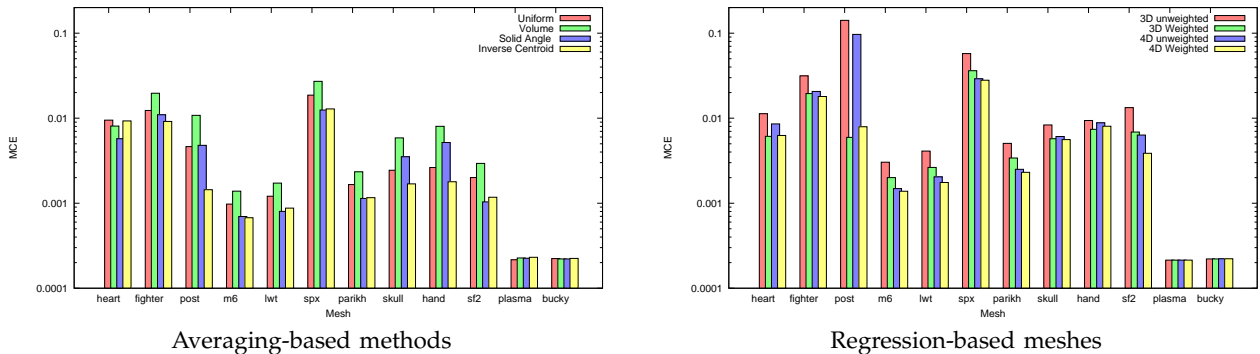Averaging-based methods   Regression-based meshes

Fig. 11. Comparison of gradient reconstructions for different meshes with an embedded spherical scalar field.
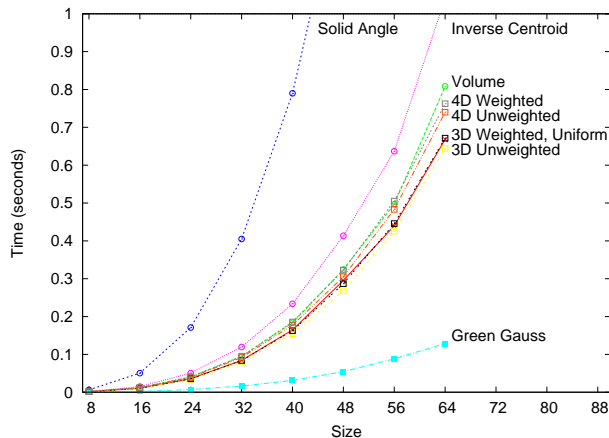
Fig. 12. Timing for test benchmarks. The Green Gauss method is significantly faster than the rest (including volume-weighted averaging, which produces equivalent results). Weighting may be costly for averaging methods, but does not have a significant impact in regression-based methods.

4.8 Efficiency Analysis

Another dimension of our evaluation is the computational cost of the different methods. The results are summarized in Fig. 12 for the synthetic data sets. We used an Intel Core 2 quad-core processor with 3.0 GHz and 4GB of RAM. The results were obtained in a CPU-based implementation using a comparable set of operations for each of the methods. We expect the relative performance of each method to be exhibited under different machines, including GPUs. Although parallel computation may decrease the gap between the different methods, the relative cost per element is still the same. Cell average is in general the costliest approach, in particular for the solid angle and inverse centroid weighting schemes. Green Gauss, on the other hand, is the fastest of all methods. Compare it to averaging with volume weighting, which provides equivalent results. Regression methods do not differ greatly in speed, and the cost of weighting is insignificant in comparison to unweighted regression, and are out-weighted by their benefits in accuracy. Due to the speed, Green Gauss reconstruction is attractive for interactive visualization. However, as seen above, volume weighted reconstruction is not robust to low quality meshes, whereas 3D weighted regression is. This motivates one of our heuristic methods, which combines the benefits of these two approaches.

4.9 Visual Comparison

Fig. 13 shows the volume rendering of a jet wing wind simulation data set. This is a representative mesh of CFD simulations, where an adaptive mesh is used to populate densely regions of interest (around the wing and missile). In our rendering, we aim at highlighting the shock waves on the wing. Lighting helps understand the shape of these shock waves. As suggested by the quantitative results, all methods produce good results on high-quality regions, while differences become noticeable for poor quality tetrahedra. As we move from averaging methods to 3D regression, we see a decrease in quality. At the bottom of Fig. 13, we provide close-up views of certain regions of the volume and compare the results for inverse centroid weighted average and unweighted 3D regression. On the left, we see the effects of meshing more noticeable in the case of 3D regression

as shown in the shape of the specular highlight. In the middle, weighted average shows a smoother surface near the boundary of the mesh. On the right, a portion of one of the shock waves appears smooth when using averaging, but appears broken and bumpy for the case of 3D regression.

Fig. 14 shows a different case, where a smoothly changing function is embedded in a variable quality mesh (oxygen post data set). Both weighted regression and cell average methods produce similar results. The Green-Gauss method is also similar, except for a few changes in the apparent curvature of the shapes due to the use of low quality cells near the boundary. Unweighted regression suffers most from the low quality of the mesh. In this case, poor quality gradient reconstruction leads to the appearance of creases and folds where there is none, and the apparent flatness of regions where there should be a smooth curved surface.

5 IMPLICATIONS

Our study has provided interesting insight on the behavior of different gradient reconstruction methods for the purposes of lighting. We summarize these as follows:

(1) Averaging-based methods provide the most accurate gradient reconstruction in general, and the inverse centroid distance is consistently more accurate than other methods, such as Green Gauss, that are based on volume weighting. This is especially true for highly regular meshes.

(2) Regression-based methods are less accurate, but can adapt better to low quality meshes. Unweighted regression leads to poor gradient reconstructions, especially when the mesh is highly irregular. Since inverse distance weighted regression has a similar computational cost, it should be preferred. The benefits of this weighting, however, do not apply for 4D regression, which behaves similar to volume weighted average.

(3) To improve the accuracy of the reconstruction, one may have to increase the neighborhood. Although this would require an additional structure for weighted cell average methods, as it requires to find the extended cell neighborhood of a cell, it becomes easy to do for regression-based methods. Inverse distance weighting leads to a Moving Least Squares reconstruction, popular in surface reconstruction from point sets. Although introducing more points produces smoother results, they are not necessarily more accurate, as it approximates regions with high frequencies to planes.

(4) Although volume weighted average and Green-Gauss are equivalent, their implementations differ greatly and the speed gap becomes noticeable. The Green-Gauss method then becomes attractive for interactive rendering systems.

Based on our observations we propose two heuristics for obtaining high-quality rendering of unstructured meshes at reasonable speeds.

5.1 Hybrid Gradient Reconstruction

Similar to recent hybrid rendering methods, we can derive a hybrid gradient reconstruction method that favors a method depending on the local structure of the mesh. Many unstructured meshes, particularly in CFD simulations, are defined adaptively, so that large tetrahedra are located in regions of low interest, whereas small tetrahedra populate regions of interest. Connecting these usually results in tetrahedra of disparate size and aspect ratios. By probing the mesh quality in a pre-processing step, we can define the gradient using averaging
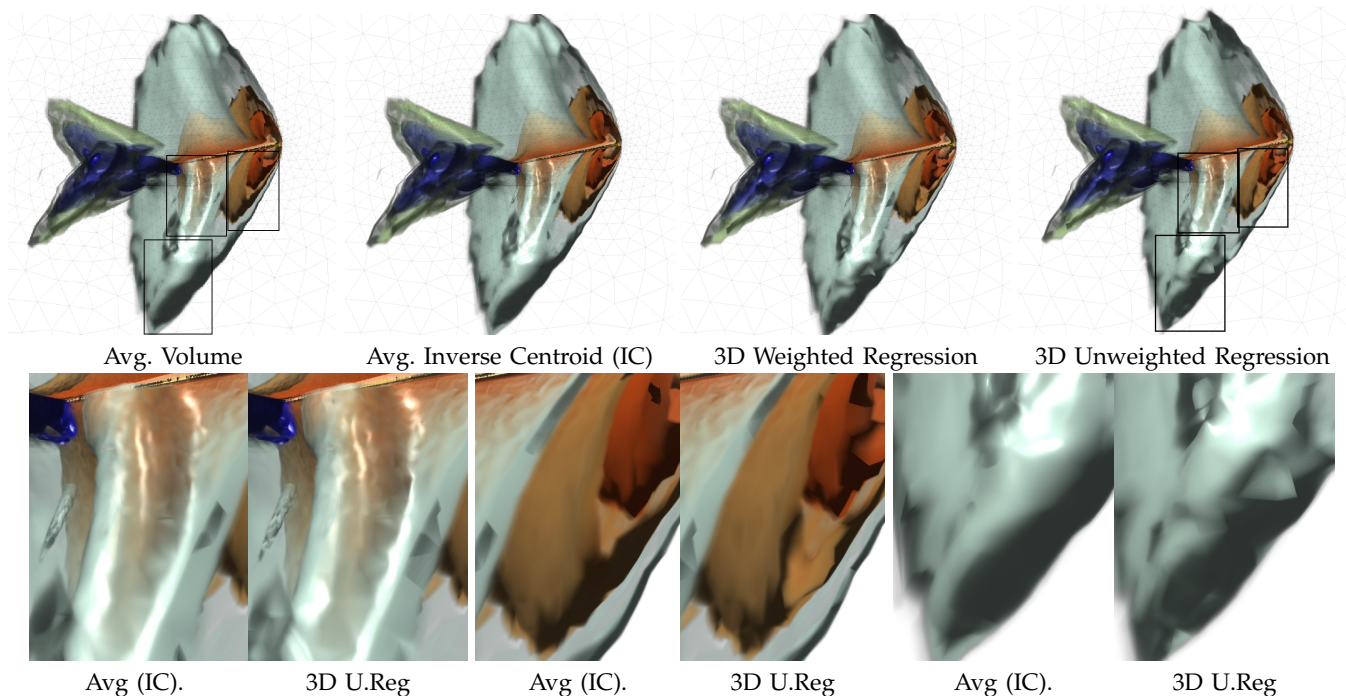
Fig. 13. Visual comparison of four methods for a turbulent flow dataset. Here, lighting helps understand the shape of the shock waves around the wing of an aircraft. At the bottom, close up views of different regions show that gradients in unweighted regression make the meshing evident, while surfaces appear smoother with averaging.
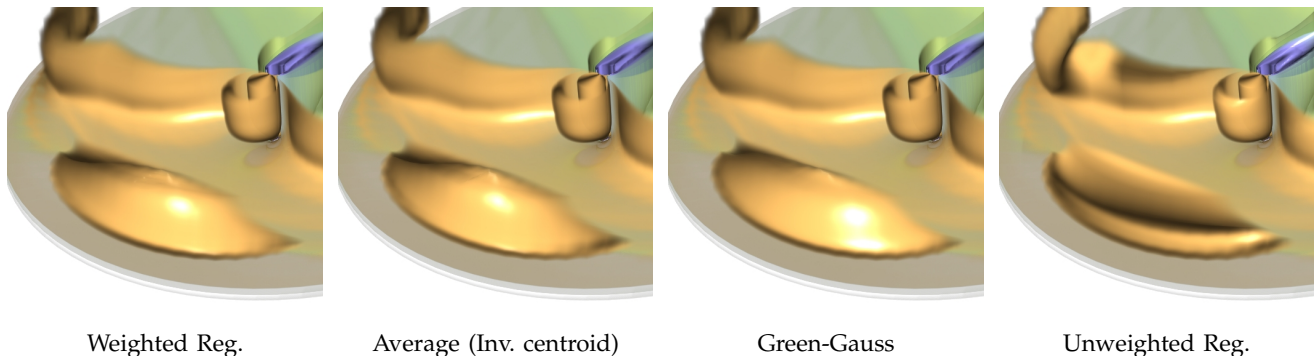

Fig. 14. Visual comparison of a smoothly changing function on a variable quality mesh (Oxygen post data set). Regions of good mesh quality (near the post) are reconstructed well for all methods. Because the function is smoothly changing, there is little difference between regression and averaging methods. For the Green Gauss methods, the shape appears more pronounced near the boundary (where there are poor quality cells). For unweighted regression, the effects of a poor quality mesh are more dramatic. Some regions appear to have creases and folds where there are none, while some other regions appear flat.

methods for high quality regions and regression for low quality regions.

For example, Fig. 15 shows the results of applying hybrid gradient reconstruction to the oxygen post data set. We see a difference between the shape of features as estimated by the weighted regression and Green Gauss methods, as suggested by the specular highlights. Green Gauss methods also introduce mesh-aligned normals near the boundaries, but the normals are, for most part, smoother. The hybrid reconstruction uses a threshold $\tau = 0.125$ on the aspect ratio to determine which method to use. Regression is only used when the minimum aspect ratio of the neighboring cells of a vertex is less than τ. This binary operator introduces discontinuities, as shown in the third image from the left. To avoid artifacts in the boundary of these two regions, we define a continuous measure, clamped between 0 and 1, and

use it as a modulation factor in linear (or spherical) blending between the two normals. Let us define $\mathbf{n}_G(x)$ and $\mathbf{n}_W(x)$ as the gradients computed using Green-Gauss and weighted regression, respectively. The normal at a given point can be found as:

$$\mathbf{n}(x) = \alpha(x)\mathbf{n}_G(x) + (1 - \alpha(x))\mathbf{n}_w(x) \qquad (20)$$

where $\alpha(x) \in [0, 1]$ is a function of volume, aspect ratio or any other metric that describes the irregularity of the mesh at any given point. To avoid computing two normals for each point, this equation only needs to compute one of them when $\alpha(x)$ tends to 0 or 1. In Fig. 15, we use:

$$\alpha(x) = \left[\frac{\min_{i \in Neigh(x)}(A(i) - \tau)}{\tau} \right]_0^1 \qquad (21)$$

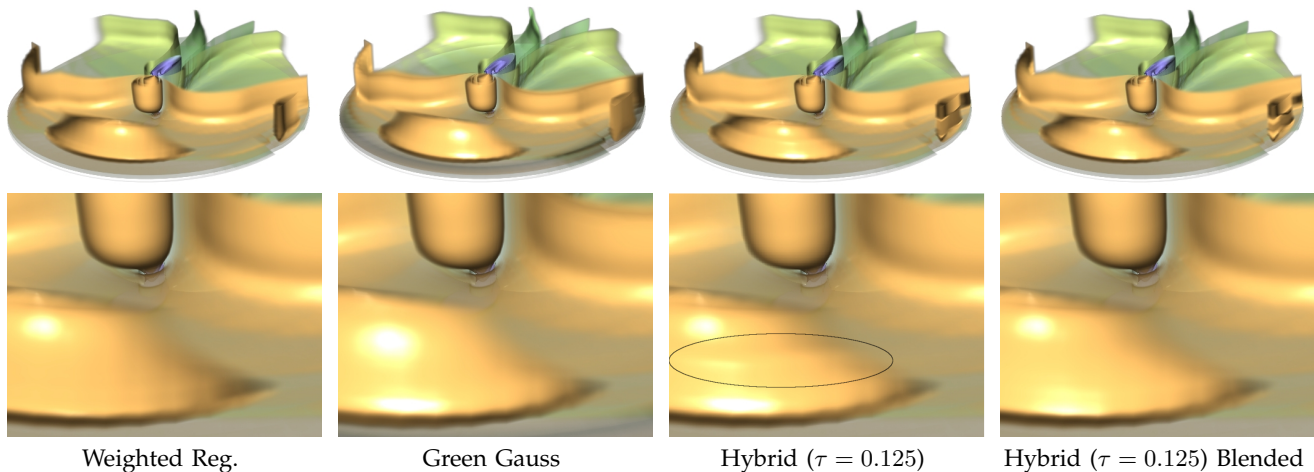where $A(i)$ is the aspect ratio of element i, $Neigh(x)$ is the set

Fig. 15. Hybrid reconstruction, based on a threshold on the aspect ratio of cells. Using a binary threshold, however (third from the left), results in visible discontinuities. With blending, the normals are adapted smoothly between both methods.

of incident cells of vertex x, and $[x]_0^1$ is a clamping operation that keeps the values between 0 and 1. Here, we assume that the aspect ratio is defined between 0 and 1, such that badly shaped elements have aspect ratio close to 0. Notice how the discontinuity is removed and the shape now appears smooth.

We tested our hybrid reconstruction in a set of meshes of variable element shape. Similar to the meshes constructed for Section 4.4, we varied the scaling factor along one of the dimensions. Instead of a uniform scaling (generating elements of same quality), we modulate the scaling along a direction, giving a range of elements from regular to irregular shapes in the same mesh. Fig. 16 shows the results for a hybrid reconstruction between Green Gauss and weighted regression using two different thresholding values. As expected, the accuracy of the hybrid reconstruction follows the "best" reconstruction for a given mesh. When the shape is highly regular, the hybrid reconstruction behaves as an averaging method, and conversely, when the shape is bad, the reconstruction behaves as a regression method. Interestingly, the hybrid reconstruction improves the mean accuracy over weighted regression for highly irregular meshes, suggesting that local adaptation of the gradient reconstruction helps reduce the error variability introduced by elements of varying shape.

5.2 Fixed-Size Neighborhood Reconstruction

One of the issues with gradient reconstruction is the reliance on connectivity information. Certain rendering mechanisms, such as cell projection, do not require connectivity information. Others, such as raycasting, use cell-to-cell connectivity to traverse rays into the mesh, at the cost of increased memory requirements. This poses a problem for dynamic meshes, where gradients need to be re-computed as the scalar function and the mesh change. We can alleviate the requirements with the use of fixed-size neighborhoods for gradient computation. For time-varying data, this fixed-size neighborhood can be pre-computed. Furthermore, it can be efficiently encoded in the texture memory of contemporary graphics cards for GPU-assisted volume rendering.

Regression methods can be easily set up to handle fixed-size neighborhoods, since they provide the best (linear) fit of the gradient function to the available set of points. The choice of neighbors, however, may also affect the quality of
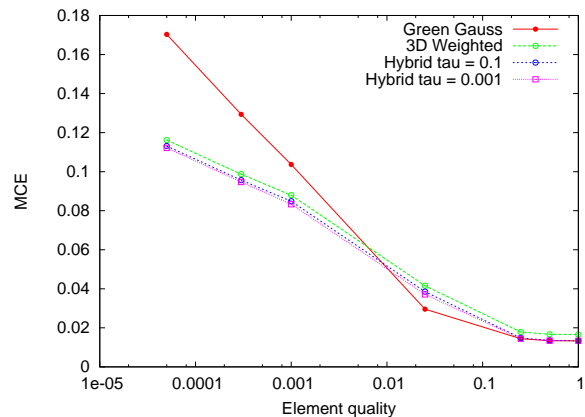

Fig. 16. Accuracy of the hybrid reconstruction scheme. For irregular elements, the hybrid method favors the regression method. As the shape quality improves, the hybrid method favors the Green Gauss method. As seen in the plot, the adaptive scheme also improves the mean accuracy.

the reconstruction. According to our evaluation, *in general*, inverse distance weighted regression provides better estimates of the gradient than unweighted regression. If we estimate the gradient based on a subset of the neighbors, the order in which we select them becomes important. Similar to meshless approximations, nearest neighbors should contribute more to the gradient. To test the effects of neighborhood size and selection criteria, we computed the gradient for an unstructured mesh under three neighbor ranking procedures: (1) Unsorted, which chooses $K < N$ neighbors randomly, (2) distance, which ranks them in decreasing order based on distance, and (3) inverse distance, which performs the inverse ranking. Fig. 17 shows the result in the mean error. We notice that inverse distance ranking provides more accurate reconstructions than other rankings. Interestingly, for only 3 neighbors, unsorted neighbors produce a better estimate. Second, we notice that inverse distance ranking results in higher accuracy than considering the full neighborhood, which suggest the presence of outliers. Similar results were obtained for other meshes of varying size and element shape.

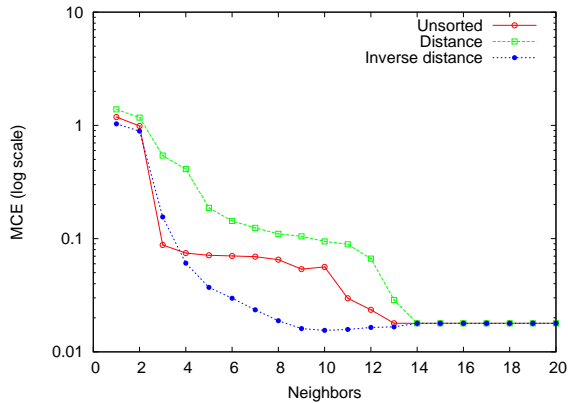An important implication is that progressive lighting of

Fig. 17. Effect of ranked fixed-size neighborhood on accuracy. We plot MCE of the estimated gradient vs. number of neighbors for three neighbor ranking procedures. Inverse distance ranking, as suggested by our quantitative analysis, produces more accurate estimates than naive (distance) or random selection.

unstructured meshes can be done on the fly. The lowest quality may be obtained by encoding only the 4 highest ranked neighbors. A medium quality can be obtained by encoding the 8 highest ranked neighbors for each vertex. Although it may not be accurate, the result is acceptable for interactive exploration. In this way, time-varying data sets can be explored at good quality without much computation cost, at the expense of extra memory required for storing the neighbors.

6 IMPLEMENTATION DETAILS

The methods described above can be implemented and deployed for all the predominant unstructured-mesh rendering techniques, including cell projection, raycasting and point-based methods. Here, we discuss implementation details for realizing these methods in GPU-based raycasting. Raycasting on the GPU is achieved by encoding the unstructured mesh in one 2D and one 3D texture. Volume rendering is obtained by traversing rays into the unstructured mesh, using the connectivity to move from a cell to the next one.

In our implementation, the x,y,z coordinates of each vertex are encoded in the color components of each texel in a 2D texture `verts`. For compact encoding, we store the scalar value of each vertex in the alpha channel. Cell information is stored in a 2D texture `cells`, where each channel encodes an index to the vertices array. Raycasting is achieved by rendering a quadrilateral of the size of the screen, and creating a ray for each pixel. Each ray is traversed in the mesh by finding the intersection with the current cell and moving to the corresponding adjacent cell. For this reason, this implementation also requires the storage of the *cell-to-cell* connectivity. The per-vertex gradient can be stored in a texture similar to the texture `verts`, in a pre-processing step. This pre-processing step makes use of additional structures, such as *vertex-to-vertex* (for the case of regression) and *vertex-to-cell* (for averaging methods) connectivity. For time-varying data sets, it is desirable to compute the per-vertex gradient on-the-fly. However, the vertex-to-vertex and vertex-to-cell connectivity are costly to encode and access via textures, due to its irregularity. Fixed-size neighborhood reconstruction, however, can be encoded efficiently in the GPU. The vertex-to-vertex connectivity can be encoded in a 2D textures, up to a fixed-size (say, 4 or 8

neighbors), similar to the cell texture. In a rendering pre-pass, we traverse in parallel all vertices in the mesh by drawing the vertex texture in a quadrilateral of the size of the texture. A shader program implements the regression method on the fixed-size neighborhood and the result is directed to the gradient texture. This method, although fast, requires enough GPU memory to store the additional gradient texture. Alternatively, regression-based reconstruction of the gradient can be included in the volume rendering shader, which increases the cost of rendering, but lessens the texture memory requirements.

7 CONCLUSIONS AND FUTURE WORK

We have provided a systematic evaluation of linear gradient reconstruction methods for unstructured-mesh volume rendering. As higher order meshes become more available and the GPU implementations begin to support mixed meshes, the need for higher order gradients will become important. Our evaluation can be followed for further comparative studies. Mesh resolution and scalar field complexities are two important factors in the accuracy of gradient reconstruction methods and their bounds can be found analytically. Our experiments confirm the quadratic trend for the linear methods. However, the shape of elements is also a factor and can lead to noticeable differences when the mesh becomes highly irregular. We notice that, despite being a good weighting scheme in general, solid angle weighting of neighboring cell gradients leads to errors when the mesh elements become almost planar. Inverse-distance weighted regression, on the other hand, produces the highest accuracy as the mesh becomes highly irregular. The implementation of these methods implies additional costs. As numerous meshes have a larger ratio of cells to vertices, storing the vertex-to-vertex connectivity may be more efficient than the vertex-to-cell connectivity, suggesting a preference for regression-based methods. Alternatives such as the Green Gauss theorem are computationally more efficient but require the vertex-to-cell connectivity. Aftosmis et al. suggested an alternative implementation that uses only vertex-vertex connectivity [3]. Nevertheless, Green Gauss methods may fail to adapt to regions of badly shaped elements. Due to the lack of ground truth, volume rendering of real meshes cannot be accurately compared except for smoothness and continuity in the reconstructed shapes. Two isosurfaces may still look smooth and yet provide different cues for shape, as exhibited in our visual comparison. In other cases, the effects of meshing are evident. As higher order simulations become available for visualization, they may be used to obtain gradients of higher accuracy to measure the error of linear methods. With our study on gradient computation, we attempt at making lighting and gradient-based rendering of arbitrary meshes a commodity, and enable high-quality rendering of unstructured-mesh volume data.

REFERENCES

[1] Visual Computing Lab. Italian National Research Council. http://vcg.isti.cnr.it/joomla/index.php, 2003.

[2] Computational Visualization Center, University of Texas at Austin. Tetrahedral model of the human heart. http://cvcweb.ices.utexas.edu/cvc/, 2008.

[3] M. Aftosmis, D. Gaitonde, and T. Tavares. Behavior of linear reconstruction techniques on unstructured meshes. *AIAA Journal*, 33(11):2038–2049, 1995.

[4] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun. Variational tetrahedral meshing. *ACM Trans. Graph.*, 24(3):617–625, 2005.

[5] W. K. Anderson and D. L. Bonhaus. An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Comput. Fluids*, 23(1):1–21, 1994.

[6] T. Apel, M. Berzins, P. Jimack, G. Kunert, A. Plaks, I. Tsukerman, and M.Walkley. Mesh shape and anisotropic elements: theory and practice. In *The Tenth Conference on The Mathematics of Finite Elements and Applications*, pages 367–376, 1999.

[7] T. J. Barth and D. C. Jerspesen. The design and application of upwind schemes on unstructured meshes. In *Proc. 27th Aerospace Sciences Meeting. Paper AIAA 89-0366*, 1989.

[8] M. J. Bentum, B. B. A. Lichtenbelt, and T. Malzbender. Frequency analysis of gradient estimators in volume rendering. *IEEE Trans. on Visualization and Computer Graphics*, 2(3):242–254, 1996.

[9] S. P. Callahan, J. L. D. Comba, M. Ikits, and M.-C. T. Silva. Hardware-assisted visibility sorting for unstructured volume rendering. *IEEE Trans. on Visualization and Computer Graphics*, 11(3):285–295, 2005.

[10] P. Cignoni, C. Montani, and R. Scopigno. Tetrahedra based volume visualization. *Mathematical Visualization*, pages 3–18, 1998.

[11] M. P. Garrity. Raytracing irregular volume data. In *VVS '90: Proc. of the 1990 workshop on Volume visualization*, pages 35–40, 1990.

[12] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH '92: Proc. of the 19th annual conference on Computer graphics and interactive techniques*, pages 71–78, 1992.

[13] C. Ledergerber, G. Guennebaud, M. Meyer, M. Bacher, and H. Pfister. Volume mls ray casting. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1539–1546, 2008.

[14] B. Lévy, G. Caumon, S. Conreaux, and X. Cavin. Circular incident edge lists: a data structure for rendering complex unstructured grids. In *Proc. IEEE Visualization '01*, pages 191–198, 2001.

[15] K.-L. Ma, J. V. Rosendale, and W. Vermeer. 3d shock wave visualization on unstructured grids. In *Proc. of the 1996 symposium on Volume visualization*, pages 87–94, 1996.

[16] S. R. Marschner and R. J. Lobb. An evaluation of reconstruction filters for volume rendering. In *Proc. IEEE Visualization '94*, pages 100–107, 1994.

[17] J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In *Proc. AIAA Computational Fluid Dynamics Conference. Paper AIAA 2003-3986*, 2003.

[18] J. Meredith and K.-L. Ma. Multiresolution view-dependent splat based volume rendering of large irregular data. In *Proc. IEEE Symp. on parallel and large-data visualization and graphics*, pages 93–99, 2001.

[19] T. H. Meyer, M. Eriksson, and R. C. Maggio. Gradient estimation from irregularly spaced data sets. *Mathematical Geology*, 33(6):693–717, 2001.

[20] T. Möller, R. Machiraju, K. Mueller, and R. Yagel. A comparison of normal estimation schemes. In *Proc. IEEE Visualization 1997*, pages 19–26., 1997.

[21] P. Muigg, M. Hadwiger, H. Doleisch, and H. Hauser. Scalable hybrid unstructured and structured grid raycasting. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1592–1599, Nov.-Dec. 2007.

[22] L. Neumann, B. Csébfalvi, A. König, and E. Gröller. Gradient estimation in volume data using 4D linear regression. In *Computer Graphics Forum*, volume 19(3), pages 351–358, 2000.

[23] D. R. O'hallaron and J. R. Shewchuk. Properties of a family of parallel finite element simulations. Technical report, Carnegie Mellon, 1996.

[24] M. Pauly, R. Keiser, L. P. Kobbelt, and M. Gross. Shape modeling with point-sampled geometry. *ACM Trans.Graph.*, 22(3):641–650, 2003.

[25] N. B. Petrovskaya. The impact of grid cell geometry on the least-squares gradient reconstruction. Technical report, Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, 2003.

[26] J. R. Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures. page 66, 2002.

[27] P. Shirley and A. Tuchman. A polygonal approximation to direct scalar volume rendering. *SIGGRAPH Comput. Graph.*, 24(5):63–70, 1990.

[28] G. Thürmer and C. A. Wüthrich. Normal computation for discrete surfaces in 3d space. *Comput. Graph. Forum*, 16(3):15–26, 1997.

[29] M. Weiler and T. Ertl. Hardware-software-balanced resampling for the interactive visualization of unstructured grids. In *Proc. IEEE Visualization 2001*, pages 199–206, 2001.

[30] M. Weiler, M. Kraus, M. Merz, and T. Ertl. Hardware-based ray casting for tetrahedral meshes. In *Proc. IEEE Visualization 2003*, pages 333–340, 2003.

[31] R. Westermann. The rendering of unstructured grids revisited. In *Proc. of the 3rd Joint IEEE TVCG-EUROGRAPHICS Symposium on Visualization (VisSym 2001)*, pages 65–74, 2001.

[32] L. Westover. Footprint evaluation for volume rendering. In *SIGGRAPH '90: Proc. of the 17th conference on Computer graphics and interactive techniques*, pages 367–376, 1990.

[33] R. Yagel, D. Cohen, and A. Kaufman. Normal estimation in 3D discrete space. *The Visual Computer*, 8(5-6):278–291, 1992.

[34] Y. Zhou and M. Garland. Interactive point-based rendering of higher-order tetrahedral data. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):1229–1236, 2006.

Carlos D. Correa received the BS degree in computer science from EAFIT University, Colombia, in 1998, and the MS and PhD degrees in electrical and computer engineering from Rutgers University in 2003 and 2007, respectively. Currently, he is a postdoctoral researcher in the Department of Computer Science, University of California, Davis. His research interests include computer graphics, visualization and user interaction.

Robert Hero is a PhD student in computer science from the University of California, Davis. He obtained his MS degree in computer science from the University of California, Santa Cruz. His research interests include GPU-based computing and visualization.

Kwan-Liu Ma received his PhD degree in computer science from the University of Utah in 1993. He is a professor of computer science at the University of California, Davis, and he directs the DOE SciDAC Institute for Ultrascale Visualization. His research interests include visualization, high-performance computing and user interface design. He received the US National Science Foundation (NSF) PECASE Award in 2000, Schlumberger Foundation Technical Award in 2001, and the UC Davis College of Engineering's Outstanding Mid-Career Research Faculty Award in 2007. He is the paper chair of the IEEE Visualization 2009 Conference. He also serves on the editorial boards of the *IEEE Computer Graphics and Applications* and the *IEEE Transactions on Visualization and Graphics*. He is a senior member of the IEEE.