

Signal-Specialized Parameterization for Piecewise Linear Reconstruction

Geetika Tewari¹, John Snyder², Pedro V. Sander³, Steven J. Gortler¹, and Hugues Hoppe²

¹ Harvard University, Cambridge, MA, USA

² Microsoft Research, Redmond, WA, USA

³ ATI Research, Marlborough, MA, USA

Abstract

We propose a metric for surface parameterization specialized to its signal that can be used to create more efficient, high-quality texture maps. Derived from Taylor expansion of signal error, our metric predicts the signal approximation error - the difference between the original surface signal and its reconstruction from the sampled texture. Unlike previous methods, our metric assumes piecewise-linear reconstruction, and thus makes a good approximation to bilinear reconstruction employed in graphics hardware. We achieve significant savings in texture area for a desired signal accuracy compared to the signal-specialized parameterization metric proposed by Sander et al. in the 2002 Eurographics Workshop on Rendering.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Color, shading, shadowing and texture

1. Introduction

Texture mapping features in graphics hardware are being increasingly used in real-time rendering. Surface signals play an important role in achieving a number of rendering effects. While these rendering effects can also be stored per vertex on a high resolution mesh, using a coarser mesh with textures is generally more efficient [COM98].

Texture mapping requires a surface to be parameterized onto a texture domain by assigning texture coordinates to its vertices. Given this parameterization, the surface signal is sampled into a texture image of a given resolution. Texture memory can become a scarce resource in complex scenes with many textured objects. In this paper we examine how to construct a parameterization to best represent a given surface signal using textures as compact as possible. We construct such a parameterization as an off-line, automatic preprocess.

The majority of surface parameterization schemes assume no *a priori* knowledge of the signal and instead minimize for various geometric distortion measures such as preservation of area and angles. Sander et al. build a surface parameterization optimized for a specific signal by trying to reduce the *signal approximation error* - the difference between the re-

constructed signal and the original signal [SGSH02]. While their approach already produces more efficient texture maps than signal-independent parameterizations, it is based on the assumption that the reconstruction is piecewise constant. The *integrated metric tensors* in their signal-specialized metric distinguish between constant and linear signals but do not differentiate between linear and higher-order signals. Given that graphics hardware employs bilinear interpolation, their metric could lead to over or undersampling, depending upon the surface signal.

In this paper, we examine minimizing the *signal approximation error* with the assumption that the reconstruction is piecewise linear. Based on Taylor polynomial expansion of signal error our metric consists of a weighted sum of squares of second derivatives of the mapping h from the texture domain (s, t) to an n dimensional surface signal. These second derivatives allow our metric to distinguish between linear and higher order signals and thus provide greater sensitivity to signal detail. This results in reduced signal approximation error for a given texture size (see Fig. 7).

The specific contributions of this paper are:

- A novel error metric that integrates signal approximation

error over the surface. It is derived using Taylor expansion of signal error with the assumption that the reconstruction is piecewise linear.

- A method to compute Hessians, second derivatives of the mapping h between the texture domain and the surface signal, using a least squares fitting method in order to evaluate the terms of the metric. These terms are precomputed and integrated over each face.
- An affine transformation rule for efficient evaluation of the metric during the parameterization process.

We implement an efficient algorithm that minimizes the signal error over the parameterization, while maintaining an embedding. Our optimization algorithm incorporates a multiresolution hierarchy [SGSH02] to propagate metric information from fine-to-coarse and creates the parameterization in a coarse-to-fine manner. In Section 6 we compare the signal approximation error of texture maps created using our metric with that of Sander et al., and show that our metric achieves significant savings in texture area for a desired signal accuracy.

2. Previous Work

Signal Independent Parameterizations The problem of minimizing distortion while flattening a surface chart into 2D has been studied in great detail since 1995 as noted in the survey by Floater and Hormann [FH04].

Pinkall and Polthier first introduce the notion of using cotangent weights as a discrete measurement of Dirichlet energy with the aim of computing minimal surfaces [PP93]. Eck et al. propose the use of Dirichlet energy minimization to parameterizing a mesh using harmonic maps [EDD*95]. The texture coordinates for boundary vertices, however, must be fixed *a priori* and harmonic maps may contain face flips (adjacent faces in texture space with opposite orientation) which violate the bijectivity of a parameterization. Duchamp et al. investigate multiresolution methods for computing harmonic maps [DCDA97]. Based on earlier work by Tutte [Tut60], Floater [Flo97] proposes a different set of weights for the edge spring model that guarantees an embedding if the texture coordinates of the boundary are fixed to a convex polygon. Desbrun et al. define a space of measures spanned by a discrete version of the Dirichlet energy, and a discrete authalic energy [DMA02].

Hormann and Greiner propose the MIPS parameterization [HG00], which maximizes the conformality of the piecewise linear mapping without demanding the mesh boundary to be mapped onto a fixed shape. Another approach to minimize angular distortion is proposed by Sheffer and de Sturler [Sds01]. They define a non-linear energy in terms of the corner angles of the mesh in texture space. Levy et al. formulate the discrete conformality problem as a quadratic minimization problem and prove the uniqueness and existence of its solution [LPRM02]. Using standard numerical conjugate

gradient solver they are able to compute least squares approximations to continuous conformal maps very efficiently without requiring fixed boundary texture coordinates.

There are some methods to compute parameterizations over a non-planar domain. Haker et al. compute conformal maps from a spherical domain onto a three dimensional surface [HAT*00]. Lee et al. use mesh simplification to parameterize a surface over a base mesh [LSS*98]. Khodakovskiy et al. employ a similar approach but with emphasis on globally smooth derivatives [KLS03]. Praun et al. introduce a robust technique for directly parameterizing a genus-zero surface onto a spherical domain employing minimization of a stretch-based measure, to reduce scale-distortion and thereby prevent undersampling [PH03]. Gu and Yau solve directly for global "flow fields" over a mesh of arbitrary genus, that can be "integrated" to obtain parameterized charts [GY03].

Few approaches explicitly optimize global area or global length distortion. Maillot et al. minimize an edge length distortion, but cannot guarantee the absence of face flips [MYV93]. They also propose an area preserving energy and combine both energies in a convex combination. Levy and Mallet use a metric that combines orthogonality and isoparametric terms [LM98]. Sander et al. minimize the average or maximum singular value of the Jacobian to prevent undersampling of the surface [SSGH01]. To optimize for a uniform sampling, Sorkine et al. minimize the maximum of the maximum singular value and the inverse of the minimal singular value, which penalizes both under- and oversampling [SCOG02].

Signal-specialized parameterizations Until recently, there has been little emphasis on exploiting knowledge of the surface signal in optimizing the parameterization.

Given an existing parameterization, Sloan et al. warp the texture domain onto itself to more evenly distribute a scalar importance field [SWB98]. Unlike importance, our metric is derived directly from signal approximation error, and is integrated over the surface. Terzopoulos and Vasilescu approximate a 2D image using a warped grid of sample values [TV91]. The warping is achieved using a dynamic simulation where grid edge weights are set according to local image content. We consider signals mapped onto surfaces in 3D, define the parameterization on a coarser, irregular mesh, and store the signal in a texture image mapped onto this mesh.

Sander et al. build a signal-specialized parameterization in a muti-grid hierarchy and minimize the signal approximation error, the error between the original surface signal and its reconstruction from the sampled texture with the assumption that the reconstruction is piecewise constant [SGSH02]. In contrast, our metric is derived assuming that our reconstruction method is piecewise linear. This more closely matches texture reconstruction hardware which does bilinear interpolation. This also holds true if the parameterization is used for remeshing since the output remesh is typically made up

of triangles which linearly interpolate the vertices. In Section 6 we demonstrate significant improvement in the quality of texture maps obtained for a desired level of signal accuracy over a variety of different models.

Balmelli et al. create space-efficient texture maps by distributing frequency content uniformly across an image using a wavelet packet decomposition technique and denoising filter [BBT02]. The frequency map captures the relative importance of different regions in the image, and causes the image to be stretched in high frequency areas, and contracted in low frequency areas. There are two main differences between our approach and that adopted by Balmelli et al. First, our metric is derived specifically to reduce reconstruction errors. It is able to differentiate directional differences in the variability of the signal, and it agrees with optimality results from approximation theory. Second, given some signal over a mesh, our method simply solves for a parametrization of the mesh, and then samples and stores that signal as a single texture. In contrast, Balmelli et al. begin with the signal represented as an input texture and with associated texture coordinates. They warp the image in order to more efficiently use the texture space and subsequently update the texture coordinates to account for the warping. However, since the warping is performed on a different grid than the triangulation of the mesh, this necessarily creates some slipping of the texture over the surface.

3. Signals over Meshes

Let the surface signal be denoted by the function $g : S \rightarrow Q$, where the signal-space Q can be vector-valued (e.g. RGB color is a 3-vector in Q).

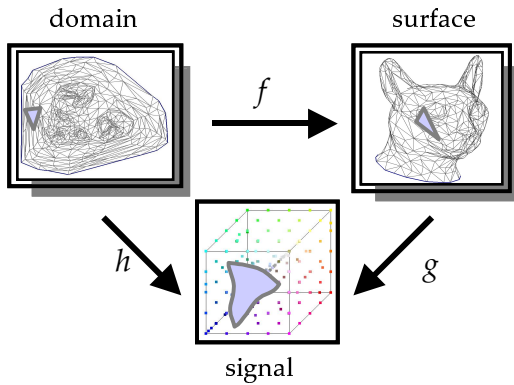


Figure 1: We examine the mapping $h = g \circ f$.

There are a number of ways a surface signal may be defined over the mesh. For example, a signal may result from the evaluation of a procedural texture computation. In this context, we wish to parameterize S and sample this signal

over the domain D . One also may begin with a signal defined by a high-resolution image-texture and a given mapping onto S . In this case we may wish to parameterize S , and resample the original signal at some lower resolution.

One can also define a signal over a low resolution mesh S , using a high resolution mesh of the same model. If attributes (e.g. RGB color, or normals) are specified at each vertex of this high resolution mesh, linear interpolation then defines a signal at each point on the high resolution mesh. *Normal shooting* [SGG*00] can then be used to create a correspondence between S and the high resolution mesh, thus defining a signal over S . All of our examples are of this type.

Normal shooting involves interpolating surface normals of the face on which a subvertex lies on the low resolution mesh S , and shooting rays in the direction of these interpolated normals to the high resolution mesh. During ray-shooting, a ray may fail to hit the high resolution model, in which case we use the nearest high resolution mesh vertex to the subvertex to estimate its signal.

4. Signal-Specialized Parameterization Metric

4.1. Metric Derivation

To find the surface parameterization f , we examine how well the function $h = g \circ f$ (from the texture domain D to the signal-space Q) is approximated when reconstructed from a discrete sampling over D (see Fig. 1).

In this section we derive a metric for signal approximation error, $E_h(s, t)$, defined as the difference between h and its reconstruction \hat{h} from a discrete sampling with spacing δ in D with the assumptions: (1) \hat{h} is a piecewise linear reconstruction, and (2) the sampling is asymptotically dense.

We assume that the domain D contains a regular grid of sample points (s_i, t_j) , spaced 2δ apart on each axis as illustrated in Fig. 2. Let $(\hat{s}, \hat{t}) \in [-\delta, +\delta] \times [-\delta, +\delta]$ be a local coordinate system within the grid square \square_{ij} about each sample, such that $(s, t) = (s_i + \hat{s}, t_j + \hat{t}) \in \square_{ij}$. Given that texture

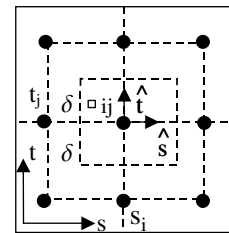


Figure 2: A regular grid of sample points in the texture domain.

mapping in hardware employs bilinear interpolation, a potential reconstruction function \hat{h} in the neighborhood \square_{ij} of

each sample is given by a function with a constant and linear term in it: $\hat{h}_{ij}(s, t) = h(s_i, t_j) + [h_s(s_i, t_j), h_t(s_i, t_j)] \begin{bmatrix} \hat{s} \\ \hat{t} \end{bmatrix}$. With this reconstruction function the pointwise squared error can be expressed as:

$$E_{ij}(s, t) = ||h(s, t) - \hat{h}_{ij}(s, t)||^2$$

$$= \left| h(s_i + \hat{s}, t_j + \hat{t}) - h(s_i, t_j) - [h_s(s_i, t_j), h_t(s_i, t_j)] \begin{bmatrix} \hat{s} \\ \hat{t} \end{bmatrix} \right|^2$$

Using a Taylor expansion about (s_i, t_j) , this can be written as

$$E_{ij}(s, t) = \hat{E}_{ij}(\hat{s}, \hat{t}) + O(\delta^5)$$

where the error term from the squared Taylor expansion gives rise to terms of order $O(\delta^5)$, which are insignificant compared to the first term of order δ^4 as $\delta \rightarrow 0$. The first term, corresponding to the third term of the Taylor expansion, is defined via

$$\hat{E}_{ij}(\hat{s}, \hat{t}) = \frac{1}{4} \left(\begin{bmatrix} \hat{s} & \hat{t} \end{bmatrix} \begin{bmatrix} h_{ss}(s_i, t_j) & h_{st}(s_i, t_j) \\ h_{st}(s_i, t_j) & h_{tt}(s_i, t_j) \end{bmatrix} \begin{bmatrix} \hat{s} \\ \hat{t} \end{bmatrix} \right)^2$$

To find the sum of squared error over all components of a vector function, we obtain

$$\hat{E}_{ij}(\hat{s}, \hat{t}) = \frac{1}{4} \sum_{k=1}^n \left(\begin{bmatrix} \hat{s} & \hat{t} \end{bmatrix} \begin{bmatrix} h_{ss}^k(s_i, t_j) & h_{st}^k(s_i, t_j) \\ h_{st}^k(s_i, t_j) & h_{tt}^k(s_i, t_j) \end{bmatrix} \begin{bmatrix} \hat{s} \\ \hat{t} \end{bmatrix} \right)^2$$

$$= \frac{1}{4} (\alpha(s_i, t_j) \hat{s}^4 + 4\beta(s_i, t_j) \hat{s}^3 \hat{t} + (4\tau(s_i, t_j) + 2\chi(s_i, t_j)) \hat{s}^2 \hat{t}^2 + 4\varepsilon(s_i, t_j) \hat{s} \hat{t}^3 + \phi(s_i, t_j) \hat{t}^4)$$

where n is the dimensionality of the signal function h (for instance $n = 3$ for RGB color), and h^k is the k -th component of h , and

$$\begin{bmatrix} \alpha(s_i, t_j) & \beta(s_i, t_j) & \chi(s_i, t_j) \\ \beta(s_i, t_j) & \tau(s_i, t_j) & \varepsilon(s_i, t_j) \\ \chi(s_i, t_j) & \varepsilon(s_i, t_j) & \phi(s_i, t_j) \end{bmatrix}$$

$$= \sum_{k=1}^n \begin{bmatrix} h_{ss}^k(s_i, t_j) \\ h_{st}^k(s_i, t_j) \\ h_{tt}^k(s_i, t_j) \end{bmatrix} [h_{ss}^k(s_i, t_j) \ h_{st}^k(s_i, t_j) \ h_{tt}^k(s_i, t_j)]$$

$$\equiv H(s, t)$$

We can integrate $\hat{E}_{ij}(\hat{s}, \hat{t})$ over $\square = [-\delta, +\delta] \times [-\delta, +\delta]$ to obtain

$$\tilde{E}_{ij}(\square) \approx \frac{\hat{A}_{ij}}{4} \delta^4 \left(\frac{\alpha}{5} + \frac{(4\tau + 2\chi)}{9} + \frac{\phi}{5} \right)$$

where \hat{A}_{ij} is the 3D surface area corresponding to this grid square. Odd powers of \hat{s} or \hat{t} vanish after integration with symmetric limits i.e., $[-\delta, +\delta]$. The total error over all grid cells is

$$\tilde{E}_\delta(S) = \sum_{ij} \frac{\hat{A}_{ij}}{4} \delta^4 \left(\frac{\alpha}{5} + \frac{(4\tau + 2\chi)}{9} + \frac{\phi}{5} \right)$$

and its limit as $\delta \rightarrow 0$ is given by

$$\lim_{\delta \rightarrow 0} (\tilde{E}_\delta(S))$$

$$= \lim_{\delta \rightarrow 0} \frac{1}{4} \delta^4 \iint_{(s,t) \in S} \left(\frac{\alpha(s,t)}{5} + \frac{(4\tau(s,t) + 2\chi(s,t))}{9} + \frac{\phi(s,t)}{5} \right) dA(s, t)$$

where $dA(s, t)$ is the differential surface area. Evidently the error converges to 0 at a rate of $O(\delta^4)$. Therefore, a measure of asymptotic approximation error with piecewise linear reconstruction over the entire surface S is to find the rate of convergence,

$$\tilde{R}(S) \equiv \lim_{\delta \rightarrow 0} \left(\frac{\tilde{E}_\delta(S)}{\delta^4} \right)$$

$$= \frac{1}{4} \iint_{(s,t) \in S} \left(\frac{\alpha(s,t)}{5} + \frac{(4\tau(s,t) + 2\chi(s,t))}{9} + \frac{\phi(s,t)}{5} \right) dA(s, t)$$

This can be expressed as:

$$\tilde{R}(S) = \frac{1}{4} \text{welsum}(\tilde{H}(S))$$

where

$$\tilde{H}(S) = \begin{bmatrix} \tilde{\alpha} & \tilde{\beta} & \tilde{\chi} \\ \tilde{\beta} & \tilde{\tau} & \tilde{\varepsilon} \\ \tilde{\chi} & \tilde{\varepsilon} & \tilde{\phi} \end{bmatrix} = \iint_{(s,t) \in S} H(s, t) d\hat{A}(s, t)$$

is the *integrated H* of the signal function h , and *welsum* (weighted element sum) is defined via

$$\text{welsum} \left(\begin{bmatrix} \tilde{\alpha} & \tilde{\beta} & \tilde{\chi} \\ \tilde{\beta} & \tilde{\tau} & \tilde{\varepsilon} \\ \tilde{\chi} & \tilde{\varepsilon} & \tilde{\phi} \end{bmatrix} \right) = \frac{1}{5} \tilde{\alpha} + \frac{2}{9} \tilde{\chi} + \frac{4}{9} \tilde{\tau} + \frac{1}{5} \tilde{\phi}$$

Reducing this integral to a sum over domain triangles, T , we obtain

$$\tilde{H}(S) = \sum_{i \in T} \tilde{H}(\Delta_i)$$

where

$$\tilde{H}(\Delta_i) = \iint_{(s,t) \in \Delta_i} H(s, t) d\hat{A}(s, t)$$

With very similar analysis it is possible to derive a measure for bilinear reconstruction. For the sake of simplicity, we chose to implement the metric for piecewise-linear reconstruction.

4.2. Properties of our Metric

The approximation theory/finite element literature provides various asymptotic results on optimal triangulations of the plane. In particular, suppose one wishes to approximate

some bivariate scalar function $g(x, y)$ using linear interpolation with a given number of triangles over the (x, y) plane. If approximation error is measured in the L^2 sense then, according to Nadler, as the number of triangles goes to infinity, an optimal triangle's orientation is given by the eigenvectors of the Hessian of g and its aspect ratio by

$$\left| \frac{\lambda_{max}}{\lambda_{min}} \right|^{\frac{1}{2}}$$

where λ_{max} and λ_{min} are eigenvalues of the Hessian of the approximation function [Nad86]. It has also been shown that a mesh best approximates a smooth surface if the anisotropy of the mesh follows (in non-hyperbolic regions) the eigenvalues and eigenvectors of the curvature tensor of the smooth surface regions [Sim94].

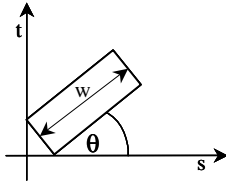


Figure 3: A rectangle R of width w rotated by θ .

It can be shown that, at least locally, our metric agrees with Nadler's result. Assume that our surface S is the unit square on the x, y plane, and that the signal is some quadratic function $g(x, y) = Ax^2 + By^2 + F$ for which $|A| > |B|$. This considers quadratics in normal form; a general quadratic can always be reduced to normal form by a translation and rotation of the x, y domain. It is easy to see that $2A$ and $2B$ are eigenvalues of the Hessian of g

$$\begin{bmatrix} 2A & 0 \\ 0 & 2B \end{bmatrix}$$

In this case, Nadler's result implies that the optimal sampling on a uniform grid will be axis aligned and have aspect ratio of

$$\left| \frac{A}{B} \right|^{\frac{1}{2}}$$

Now, assume that we parameterize S over the (s, t) texture domain, using a single affine map from S to a rectangle R of width w , height $\frac{1}{w}$, and orientation θ (see Fig. 3). Over these two degrees of freedom, (w, θ) , we minimized our definition of signal error in closed form and verified that the following are indeed local minima as predicted by [Nad86]: θ an integer multiple of $\frac{\pi}{2}$, and $w = \left| \frac{B}{A} \right|^{\frac{1}{4}}$. Thus our metric also obtains the optimal orientation (axis-aligned) and optimal aspect ratio ($w/h = w^2 = \left| \frac{B}{A} \right|^{\frac{1}{2}}$).

Nadler further identifies a one degree of freedom family of optimal solutions for the case in which the Hessian

has a negative eigenvalue (i.e, the surface is locally hyperbolic) [Nad86]. It remains for future work to show that this entire family minimizes our metric, and that our metric admits no minima outside Nadler's set.

4.3. Affine Transformation Rule for \tilde{H}

During optimization, we need to repeatedly modify the parameterizations of mesh vertices and compute the resulting change in the signal error. Recomputing \tilde{H} matrices based on the modified parameterization is undesirable since this involves expensive numerical integration as well as recomputing second derivatives of h .

Since modification of the parameterization is simply an affine transformation of each mesh triangle, we can exactly compute \tilde{H} of a transformed triangle from its original H without having to do the computation explicitly.

Let $e : D \rightarrow D : (s', t') \rightarrow (s, t)$ be a local affine transform from the new triangle parameterization to the old, resulting in the new map $h' = h \circ e$. Let $J(s, t)$ be the Jacobian of the mapping e

$$J(s, t) = \begin{bmatrix} \frac{\partial e_1}{\partial s} & \frac{\partial e_1}{\partial t} \\ \frac{\partial e_2}{\partial s} & \frac{\partial e_2}{\partial t} \end{bmatrix} = \begin{bmatrix} p(s, t) & q(s, t) \\ r(s, t) & s(s, t) \end{bmatrix}$$

where e_1 denotes the first coordinate of e and e_2 the second and where the coordinates (s, t) now parameterize the domain of e and thus h' . Thus the transformed second derivatives of a scalar function h' are given by

$$\begin{aligned} \begin{bmatrix} h'_{ss} & h'_{st} \\ h'_{st} & h'_{tt} \end{bmatrix} &= J^T \begin{bmatrix} h_{ss} & h_{st} \\ h_{st} & h_{tt} \end{bmatrix} J \\ &= \begin{bmatrix} p(s, t) & r(s, t) \\ q(s, t) & s(s, t) \end{bmatrix} \begin{bmatrix} h_{ss} & h_{st} \\ h_{st} & h_{tt} \end{bmatrix} \begin{bmatrix} p(s, t) & q(s, t) \\ r(s, t) & s(s, t) \end{bmatrix} \end{aligned}$$

Expressing this as a linear system in the untransformed second derivatives:

$$\begin{bmatrix} h'_{ss} \\ h'_{st} \\ h'_{tt} \end{bmatrix} = \begin{bmatrix} p^2 & 2pr & r^2 \\ pq & (ps+qr) & rs \\ q^2 & 2qs & s^2 \end{bmatrix} \begin{bmatrix} h_{ss} \\ h_{st} \\ h_{tt} \end{bmatrix} = Q \begin{bmatrix} h_{ss} \\ h_{st} \\ h_{tt} \end{bmatrix}$$

Thus H can be transformed via

$$Q \begin{bmatrix} \alpha & \beta & \chi \\ \beta & \tau & \epsilon \\ \chi & \epsilon & \phi \end{bmatrix} Q^T = Q \begin{bmatrix} h_{ss}^i \\ h_{st}^i \\ h_{tt}^i \end{bmatrix} [h_{ss}^i \ h_{st}^i \ h_{tt}^i] Q^T$$

yielding $H' = QHQ^T$. In the case of a vector function h , we can absorb the sum over components inside the H term. Integrating, we obtain:

$$\tilde{H}'(\delta_i) = \iint_{(s,t) \in A} QHQ^T d\hat{A}(s, t)$$

where Q is a function of the Jacobian as defined above, and δ_i is the i th triangle over the surface. Since the Jacobian is constant within each triangle δ_i , we can factor out

the multiplication by Q in the above integral yielding a transformation rule for $\tilde{H}(S)$

$$\tilde{H}'(S) = Q\tilde{H}(S)Q^T$$

Therefore

$$\tilde{R}'(S) = \frac{1}{4} \text{welsum}(\tilde{H}'(S))$$

4.4. Numerical Computation of \tilde{H}

We described how to apply the affine transformation rule on the \tilde{H} matrices to quickly evaluate the signal error metric. However, in order to use the affine transformation rule during the parameterization process, we need to have initial \tilde{H} matrices stored on the original mesh faces.

At the outset, we do not have a parameterization of the original mesh that we could use to compute the initial \tilde{H} matrices. Thus, for each face, we create a *canonical parameterization* based on a local *isometric flattening* of that face and its three neighboring faces. As described below, we will also need to sample these neighboring faces in order to accurately compute the \tilde{H} matrices.

During the optimization process, the affine transformation rule is used to transform the \tilde{H} matrices from their canonical parameterization to the current parameterization that is being evaluated. In this section we describe how to pre-compute \tilde{H} matrices on the original mesh faces with respect to canonical face parameterizations.

We assume that a continuous signal (e.g. normal map as shown in Fig. 4) exists over the surface. Our method of computing $\tilde{H}(\Delta_i)$, is based on a *numerical integration* approach. Specifically we apply a number of regular 1-to-4 subdivisions to a face and obtain *subfaces* and *subvertices*. We compute H at all the subfaces, sum all of these up, and multiply by the geometric area of the face, in order to obtain $\tilde{H}(\Delta_i)$. The level of subdivision depends on the detail of the mesh. For all examples in this paper, we subdivided each face into 64 subfaces (3 subdivisions) to compute H .

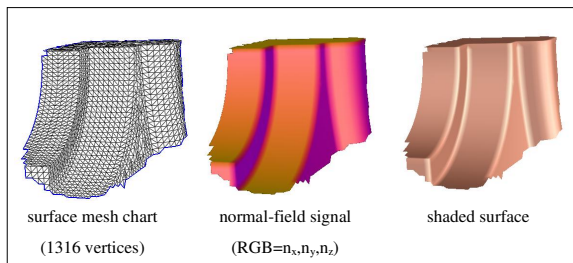


Figure 4: Our input consists of surface mesh and an associated surface signal. Here the input is a single chart of the fandisk mesh where the normal (n_x, n_y, n_z) is the signal.

To compute H at a subface, we sample the signal at some number of neighboring subvertices. In our examples, we selected the 15 closest subvertices after the 3 regular 1-to-4 subdivisions. We then estimate the second derivatives of the mapping h by finding the best fitting quadratic signal to these samples (the coefficients A, B, C, D, E and F of the quadratic $f = As^2 + Bt^2 + Cst + Ds + Et + F$). We then use the second derivatives of this quadratic approximation.

It is possible for our signals to vary linearly within two adjacent mesh faces but have dramatically different derivatives across the shared edge. Unless the subvertices used to compute the second derivatives of h over a face include subvertices from adjacent faces, the second derivatives estimated by our Least Squares solver along the boundary of the face will be inaccurate. This could lead to high signal approximation error. In order to prevent this problem, as described above, we also include subvertices from the three neighboring faces using the face's local isometric parameterization. During numerical integration, when selecting the 15 closest subvertices, we also allow these subvertices to come from these neighboring faces.

5. Chart parameterization algorithm

The goal of the optimization procedure is to minimize signal error over the parameterizations of the mesh vertices, while maintaining an embedding. Our optimization algorithm is based on that of Sander et al. [SSGH01]. For completeness we summarize this algorithm.

After obtaining an initial parameterization, we minimize the signal error by repeatedly perturbing each vertex within the kernel of the polygon formed by its neighboring vertices. To improve the speed and result of the optimizations we adopt the multiresolution optimization scheme adapted by Sander et al. [SGSH02]. We use a multiresolution progressive mesh [Hop96] sequence to propagate \tilde{H} fine-to-coarse (FTC) from the original mesh to all coarser meshes and apply a coarse-to-fine (CTF) parameterization algorithm that uses these \tilde{H} s. During FTC we redistribute the \tilde{H} according to Sander et al.'s scheme of redistributing *integrated metric tensors (IMTs)* [SGSH02]. We use a CTF optimization with the geometric stretch metric [SSGH01] to obtain an initial parameterization in order to transform the \tilde{H} s on the finest level mesh, and thus bootstrap the iterative optimization process.

This parameterization algorithm allows chart boundary vertices to move in the texture domain at all levels of the CTF optimization algorithm. Our error metric is not scale invariant, and thus the signal error could go to zero as the chart becomes infinitely large. We achieve scale-invariance by multiplying signal error by the square of the total chart area.

The high-level algorithm can be summarized as:

```

function OptimizeChartParameterization
  Pre-compute canonical  $\tilde{H}$ s on fine mesh faces
  Construct progressive mesh of chart
  //Create an initial parameterization
  Do CTF using geometric-stretch.
  //Iteratively optimize using error metric:
  for  $n$  iterations
    Transform fine mesh  $\tilde{H}$ s using current param.
    FTC propagate  $\tilde{H}$ s to all PM meshes.
    CTF optimize signal error using  $\tilde{H}$ s.

```

6. Results

We have created signal-specialized parameterizations for several models and compared them both quantitatively and qualitatively against the signal-specialized parameterization metric of Sander et al. The comparisons are shown in Fig. 7- Fig 11. All models originated from 3D scanning. The signals on the parasaur, fandisk, and cat are per-vertex normals on a high resolution mesh, mapped to a low resolution mesh using normal-shooting. The signal on the face in Fig. 11 is defined using per vertex color data on a high resolution mesh, mapped to a low resolution mesh using normal-shooting. For the fandisk model in Fig. 9 we manually partitioned the mesh into 4 charts. Table 1 shows a comparison between the running time of our metric with that of Sander et al. on several models. Our parameterization scheme takes a few minutes to run per model.

To quantify parameterization quality, we measure signal approximation error (SAE) as rms difference on a dense set of surface points, distributed uniformly according to surface area. For each point we compute the difference between the original signal and the bilinear interpolation of the four adjacent texture samples. For vector-valued signals we use the L_2 norm.

In Fig. 5- 6 we compare the reduction in SAE by our metric with that of Sander et al. We refer to the former as signal error with piecewise linear reconstruction, and the latter as signal error with piecewise constant reconstruction. The graphs show the signal error as a function of the number of texture samples for two parameterizations.

To prevent the formation of degenerate triangles when the signal is locally constant on a region of the surface, Sander

Table 1: A comparison of running time in seconds.

	Model	#Vertices	#Charts	Sander et al.	Ours
Figure 8	Parasaurhead	3,800	1	50	100
Figure 11	Face	5,000	1	120	200
Figure 10	Cat	1,000	1	10	25
Figure 9	Fandisk	6,475	4	170	210

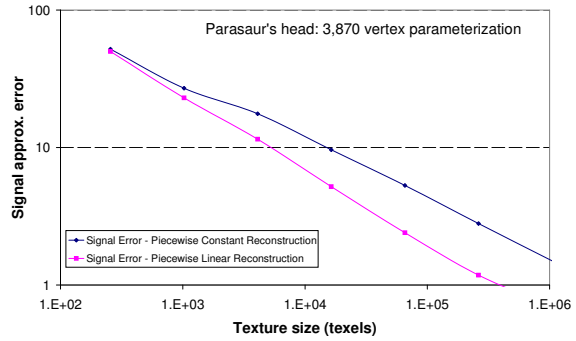


Figure 5: Comparison of signal approximation error (SAE) as a function of texture size for two parameterizations of the parasaur's head model (3,870-vertex mesh).

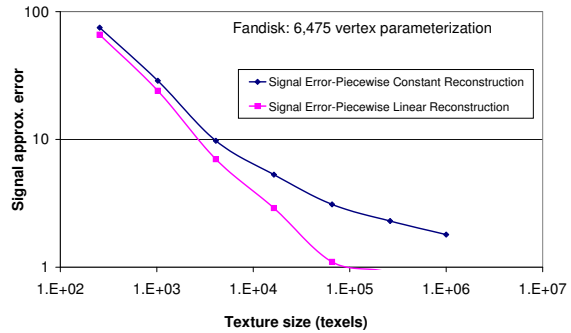


Figure 6: Comparison of signal approximation error (SAE) as a function of texture size for two parameterizations of the fandisk model (6,475-vertex mesh).

et al. add a tiny fraction of geometric stretch to their energy functional[SGSH02]. For our metric we find it necessary to do this as well on models such as the fandisk. Empirical analysis shows that the fraction of geometric stretch can have a significant impact in reducing the SAE of the parameterization. Furthermore, it seems that the amount of geometric stretch needed is dependent upon the scale of the error functional and varies from model to model. Therefore, in order to ensure fairness in our experimental comparisons in Fig. 6, we only add sufficient geometric stretch to allow the parameterization algorithm to converge to an optimal solution for both metrics. Empirical analysis also shows that for each model there is some value of the fraction of geometric stretch that produces a parameterization with the lowest SAE. For experiments with the models in Fig. 10-Fig. 11 we compare parameterizations created by our metric and the metric by Sander et al. combined with their optimal geometric stretch value that is predetermined experimentally.

Both graphs in Fig. 5- 6 show a significant reduction in error from the signal error with piecewise constant reconstruction

tion to the signal error with piecewise linear reconstruction. In particular Fig. 5 shows that using our metric, a given signal approximation error can be obtained with up to a factor of 4 savings in texture size. Fig. 7 illustrates the parameterizations of the two metrics for the parasaur model used in Fig. 5 for specific texture resolutions. Fig. 6 compares the results of the two metrics on the fandisk model and also shows savings in texture size for a given level of signal approximation error. The textures and texture mapped fandisk corresponding to a texture resolution of 128x128 are illustrated in Fig. 9. Fig. 10 and 11 are additional examples that illustrate the difference between our metric and that of Sander et al.

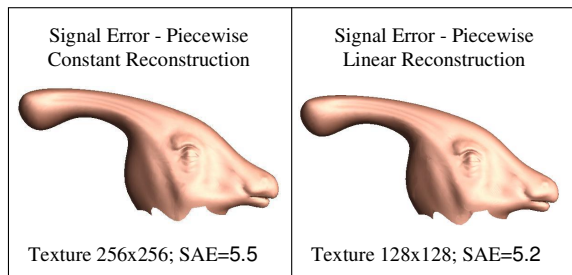


Figure 7: Our parameterization algorithm can reduce texture size by more than a factor of 4.

7. Summary and future work

Motivated by the need to reduce the size of texture maps used in real-time rendering systems, we introduced a parameterization metric that is derived from a Taylor expansion of signal approximation error. Unlike former methods, ours assumes piecewise linear reconstruction and is thus a good approximation to bilinear reconstruction employed in graphics hardware. In order to empirically evaluate our metric, we have implemented a multiresolution parameterization algorithm that minimizes the signal error defined by our metric.

Our signal-specialized parameterization metric allocates more texture samples to mesh regions with greater signal variation. In particular, the *integrated \tilde{H}* term in our metric allows it to distinguish between linear and higher order signals and allocate more texels to the latter than the former. As shown in Fig. 5- 6, a factor of 4 of savings in texture space is possible using our signal-specialized parameterization method.

There are several areas of related work that could be explored in the future. Our metric does not place any restriction on the dimensionality of the signal. It is possible to specialize a parameterization to a combination of signals, such as normals and colors. Another area for future work is to explore perceptual measures in addition to signal approximation error and propagate these measures through the rendering process.

References

- [BBT02] BALMELLI L., BERNARDINI F., TAUBIN G.: Space-optimized texture maps. *Computer Graphics Forum (Eurographics) 21*, 3 (2002), 411–420. 3
- [COM98] COHEN J., OLANO M., MANOCHA D.: Appearance-preserving simplification. In *SIGGRAPH (1998)*, pp. 115–122. 1
- [DCDA97] DUCHAMP T., CERTAIN A., DEROSE T., AND W.: *Hierarchical Computation of PL harmonic Embeddings*. Tech. rep., University of Washington., 1997. 2
- [DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. *Computer Graphics Forum (Eurographics) 21*, 3 (2002), 209–218. 2
- [EDD*95] ECK M., DEROSE T., DUCHAMP T., HOPPE H., LOUNSBERY M., STUETZLE W.: Multiresolution analysis of arbitrary meshes. In *SIGGRAPH (1995)*, pp. 173–182. 2
- [FH04] FLOATER M. S., HORMANN K.: Surface parameterization: a tutorial and survey. In *Advances on Multiresolution in Geometric Modelling*, Dodgson N., Floater M. S., Sabin M., (Eds.). Springer-Verlag, Heidelberg, Heidelberg, Denmark, 2004. 2
- [Flo97] FLOATER M. S.: Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design 14*, 4 (1997), 231–250. 2
- [GY03] GU X., YAU S.-T.: Global conformal surface parameterization. In *1st Symposium on Geometry Processing (2003)*, pp. 127–137. 2
- [HAT*00] HAKER S., ANGENENT S., TANNENBAUM A., KIKINIS R., SAPIRO G., HALLE M.: Conformal surface parameterization for texture mapping. In *IEEE TVCG (2000)*, vol. 6, pp. 181–189. 2
- [HG00] HORMANN K., GREINER G.: MIPS: An efficient global parametrization method. In *Curve and Surface Design: Saint-Malo 1999*, Laurent P.-J., Sablonnière P., Schumaker L. L., (Eds.). Vanderbilt University Press, Nashville, TN, 2000, pp. 153–162. 2
- [Hop96] HOPPE H.: Progressive meshes. In *SIGGRAPH (1996)*, pp. 99–108. 6
- [KLS03] KHODAKOVSKY A., LITKE N., SCHRODER P.: Globally smooth parameterizations with low distortion. In *SIGGRAPH (2003)*, pp. 350–357. 2

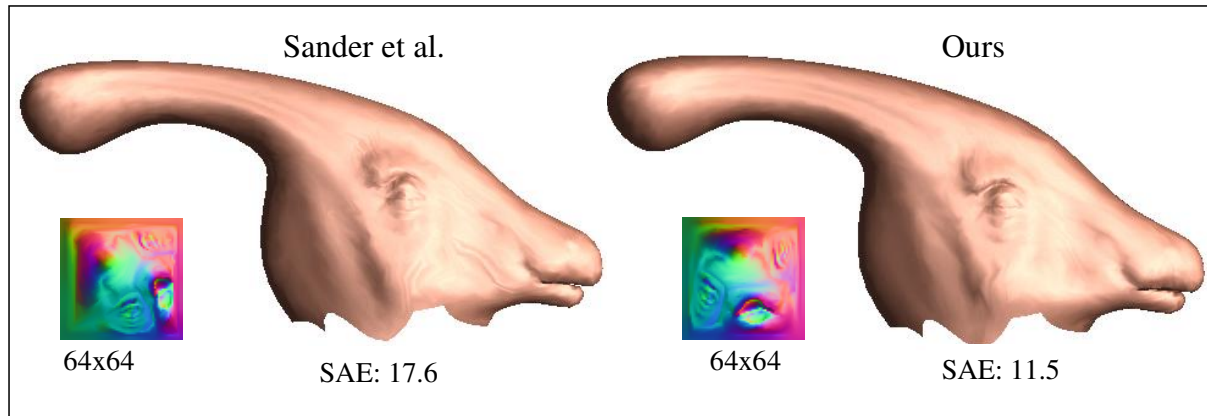


Figure 8: Texture mapped parasaur's head with 3,870 vertices.

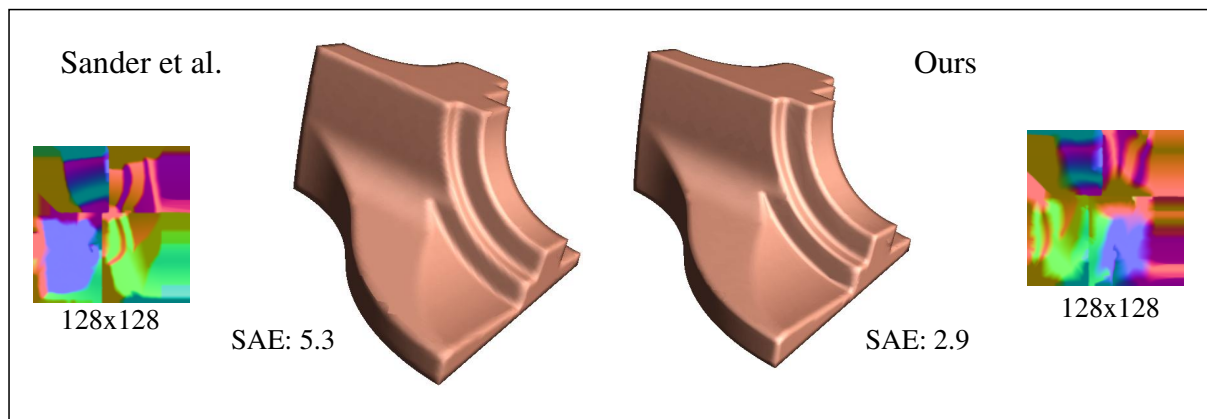


Figure 9: Texture mapped fandisk with 6,475 vertices. A "Voronoi dilation" is performed on each texture image to fill in the empty regions between the charts, and thus reduce artifacts along chart boundaries on the texture mapped model.

- [LM98] LEVY B., MALLET J.-L.: Non-distorted texture mapping for sheared triangulated meshes. In *SIGGRAPH* (1998), pp. 343–352. 2
- [LPRM02] LEVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. In *SIGGRAPH* (2002), pp. 362–371. 2
- [LSS*98] LEE A. W., SWELDENS W., SCHRODER P., COWSAR L., DOBKIN D.: Maps: Multiresolution adaptive parameterization of surfaces. In *SIGGRAPH* (1998), pp. 95–104. 2
- [MYV93] MAILLOT J., YAHIA H., VERRAUST A.: Interactive texture mapping. In *SIGGRAPH* (1993), pp. 27–34. 2
- [Nad86] NADLER E.: Piecewise linear best l_2 approximation on triangulations. In *Approximation Theory V*, et al. C. K. C., (Ed.). Academic Press, 1986, pp. 499–502. 5
- [PH03] PRAUN E., HOPPE H.: Spherical parameterization and remeshing. In *SIGGRAPH* (2003), pp. 340–349. 2
- [PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experiment. Math.* 2, 1 (1993), 15–36. 2
- [SCOGL02] SORKINE O., COHEN-OR D., GOLDENTHAL R., LISCHINSKI D.: Bounded-distortion piecewise mesh parameterization. In *IEEE Visualization* (2002). 2
- [SdS01] SHEFFER A., DE STURLER E.: Parameterization of faceted surfaces for meshing using angle-based flattening. *Engineering and Computers* 17, 3 (2001), 326–337. 2

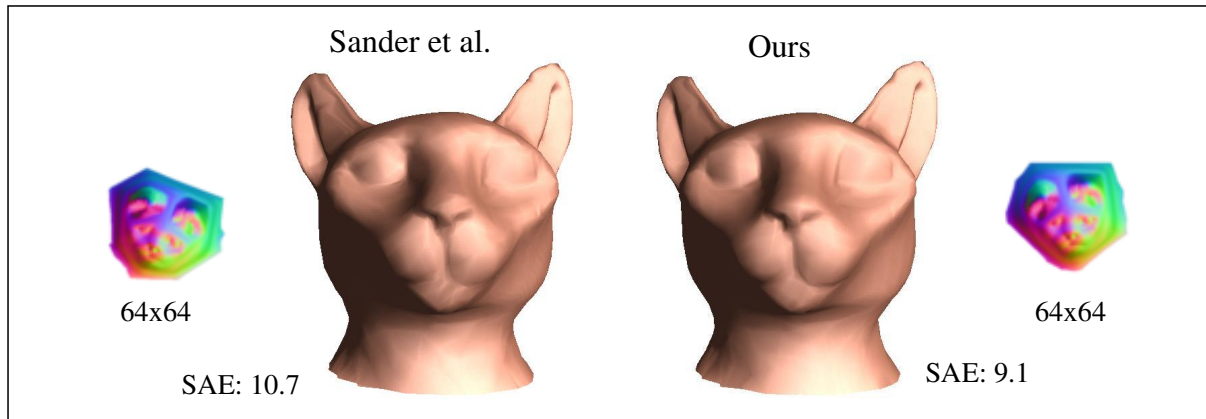


Figure 10: Texture mapped cat model with 1,000 vertices.

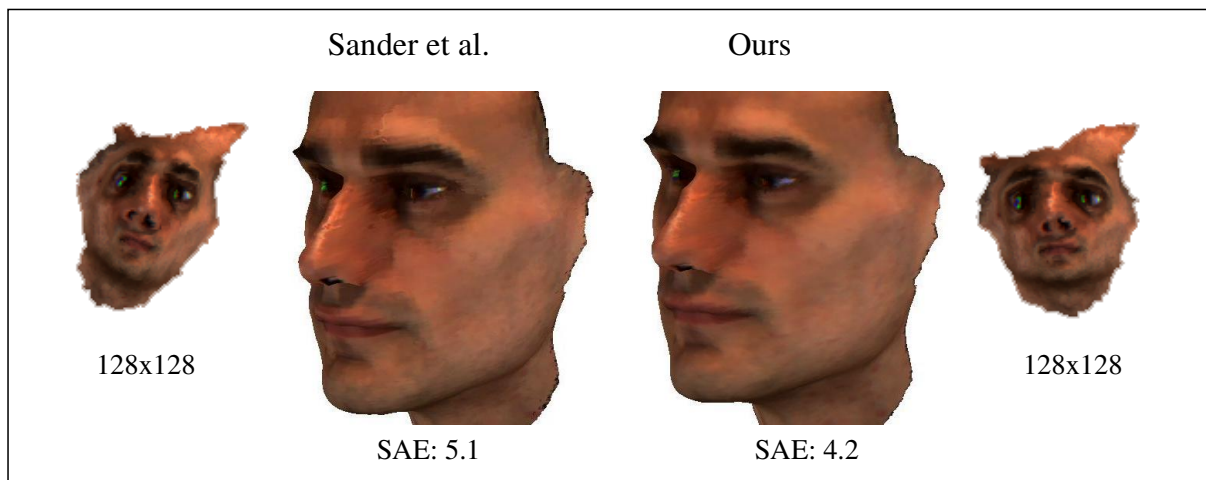


Figure 11: Texture mapped face with 5,000 vertices.

- | | |
|--|---|
| <p>[SGG*00] SANDER P. V., GU X., GORTLER S. J., HOPPE H., SNYDER J.: Silhouette clipping. In <i>SIGGRAPH</i> (2000), pp. 327–334. 3</p> <p>[SGSH02] SANDER P., GORTLER S., SNYDER J., HOPPE H.: Signal-specialized parameterization. In <i>Eurographics Workshop on Rendering</i> (2002), pp. 409–416. 1, 2, 6, 7</p> <p>[Sim94] SIMPSON R. B.: Anisotropic mesh transformations and optimal error control. <i>Appl. Num. Math</i> 14, 1-3 (1994), 183–198. 5</p> <p>[SSGH01] SANDER P., SNYDER J., GORTLER S., HOPPE H.: Texture mapping progressive meshes. In <i>SIGGRAPH</i> (2001), pp. 409–416. 2, 6</p> <p>[SWB98] SLOAN P.-P., WEINSTEIN D., BREDESON.</p> | <p>J.: Importance driven texture coordinate optimization. <i>Computer Graphics Forum (Eurographics)</i> 17, 3 (1998), 97–104. 2</p> <p>[Tut60] TUTTE W. T.: Convex representations of graphs. <i>London Math. Soc.</i> 10 (1960), 204–320. 2</p> <p>[TV91] TERZOPOULOS D., VASILESCU M.: Sampling and reconstruction with adaptive meshes. <i>CVPR</i> (1991), 70–75. 2</p> |
|--|---|