Fitting Subdivision Surfaces

Nathan Litke — Caltech
Adi Levin — Tel Aviv University
Peter Schröder — Caltech

Abstract

We introduce a new algorithm for fitting a Catmull-Clark subdivision surface to a given shape within a prescribed tolerance, based on the method of *quasi-interpolation*. The fitting algorithm is fast, local and scales well since it does not require the solution of linear systems. Its convergence rate is optimal for regular meshes and our experiments show that it behaves very well for irregular meshes. We demonstrate the power and versatility of our method with examples from interactive modeling, surface fitting, and scientific visualization.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

Keywords: Animation, CAD, Curves & Surfaces, Geometric Modeling, Digital Geometry Processing, Subdivision Schemes, Approximation, Quasi-Interpolation, Catmull-Clark

1 Introduction

Subdivision schemes provide efficient algorithms for the design, representation and processing of arbitrary topology smooth surfaces [17]. Application settings range from industrial design and animation to scientific visualization and simulation. A basic technique in any of these areas is the approximation of a desired surface with a given surface. For example, scattered data interpolation over arbitrary topology surfaces, approximation of complex iso-surfaces arising in scientific visualization, or interactive freeform surface editing. In particular the latter requires fast methods which go beyond traditional control point manipulation and its sensitivity to iso-parameter line and patch boundary layout.

In this paper we introduce a new method for fitting a subdivision surface with details to a given shape. It generalizes the method of *quasi-interpolation* to Catmull-Clark surfaces and enables efficient approximation of a given shape to *any* desired accuracy. The basic idea is to use local, weighted averages of samples of the desired surface as control points. Detail coefficients (in the sense of a Laplacian pyramid construction) are computed in a coarse to fine process in the same way, and added only where the fit is not satisfactory. In particular, no solution of global linear systems with an a priori fixed set of coefficients is required [10]. Quasi-interpolation enjoys an optimal order of convergence away from irregular vertices, making it asymptotically as good as least-squares. Empirical tests show that our method also performs well in the non-asymptotic regime.

Quasi-interpolation is useful for many data fitting tasks, particularly when the level of detail varies over the data set. Objects such as these, indicative of geometric modeling and scientific computing applications, are shown in Figure 1. Each shape is a subdivision surface with details computed using quasi-interpolation, whose fidelity to the actual object can be controlled to any desired precision. The final representation is amenable to further manipulation by performing successive local approximations on the same surface.

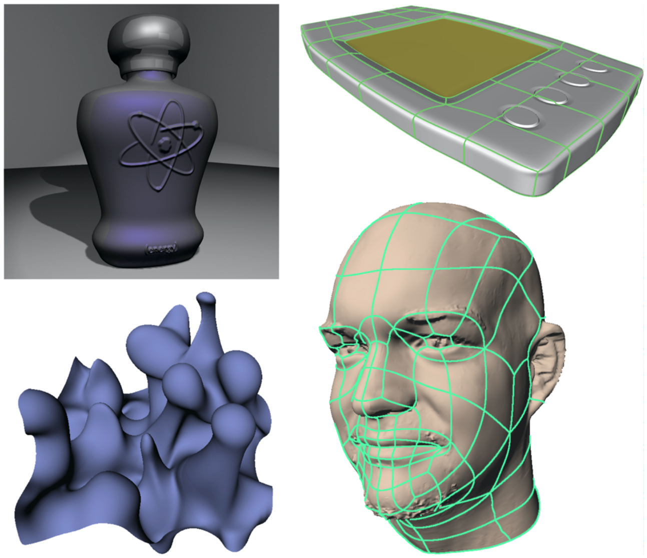

Figure 1: *Fitting applied to examples from scientific computing and geometric modeling. Each shape is a subdivision surface with details computed by quasi-interpolation.*

Note that we do not address the correspondence problem. Instead we assume that a correspondence exists, i.e., a mapping from the given surface to the desired surface, and focus on the rapid computation of appropriate, adaptively chosen detail vectors to ensure that a given error tolerance is met. In many settings such correspondences exist naturally; for example, in direct manipulation during interactive editing or in the evolution of an iso-surface in level set methods for scientific computing. In other settings establishing such mappings requires the full power of a parameterization algorithm (e.g., [13]).

Related Work Fitting subdivision surfaces with details to existing data was the subject of work by Lee et al. [12] who used a subdivision surface with displacement maps (i.e., single level details) to fit scanned geometry using a costly global optimization method. Earlier, Halstead et al. [10] described fitting of Catmull-Clark surfaces to a set of interpolation constraints. Neither approach scales well, due to the size of the linear systems and their poor conditioning, nor do they allow for *local* fitting. For complex fitting tasks, such as an emboss or an iso-surface extraction, this quickly becomes untenable. In contrast our approach is based on local operations. Thus there is no linear system to solve, providing both good scalability *and* local control.

Manipulation of subdivision surfaces in the editing setting has been studied, e.g., by Zorin et al. [18], who described hierarchical editing of Loop surfaces with details, while DeRose et al. [9] focused on Catmull-Clark surfaces and (semi-)sharp crease rules to achieve a number of modeling effects. Both rely on control point manipulation, requiring careful layout of patch boundaries and iso-parameter lines. Placing arbitrarily shaped, curve-like features on a Loop surface with details was addressed in [11] by fitting fine level control points to a given profile, but without any accuracy guarantees. Similarly, in the context of classical and hierarchical spline methods, Conrad and Mann have shown how to use quasi-interpolation for *surface pasting* [2] to achieve arbitrary placement of features in interactive editing. Unfortunately their method achieves only approximate continuity across pasting features. In contrast we extend quasi-interpolation to subdivision surfaces with details, which allows for a uniform treatment of local displacements throughout the hierarchy and guarantees globally smooth results.

The generic problem of approximation of functions by subdivision surfaces *in the regular setting* is a subject of study in approximation theory [4, 5, 6, 7, 8] where it is shown that quasi-interpolation is asymptotically as good as least squares approximation. However, in the presence of irregular vertices, optimal approximation methods are not yet known. In fact, even the optimal rate of approximation remains unknown [14]. Our algorithm generalizes quasi-interpolation to Catmull-Clark surfaces (we use the variant of Biermann et al. [1]) and extends earlier ideas applied in the context of trimming for Loop surfaces [15]. While there is currently no approximation theory for the irregular setting we find empirically that the numerical accuracy of our method compares very favorably with least-squares solutions.

2 Quasi-Interpolation

Before we review some basic facts about quasi-interpolation for surfaces, we begin with a simple univariate example to explain the difference between least-squares optimal approximation and quasi-interpolation.

An Example Consider the univariate approximation problem with uniform cubic B-splines. Given samples of a (sufficiently smooth) function f, the least-squares method computes control points for an optimal cubic spline with knots at the integers. This requires the solution of a linear system whose size is proportional to the number of knots. The approximation error can be reduced by increasing the number of knots.

Instead, the following local method can be used. Let p_i be the control point at i and set

$$p_i = -\frac{1}{6}f(i-1) + \frac{4}{3}f(i) - \frac{1}{6}f(i+1), \quad \forall i \in \mathbb{Z}. \qquad (1)$$

This is a local operator, therefore it cannot guarantee interpolation. However, this quasi-interpolant is an *exact interpolant whenever f is a cubic polynomial*. When knots (and samples) are taken at denser intervals, i.e., the spacing h is decreased, the approximation error converges to zero as $O(h^4)$, *the same rate as for least-squares approximation*. In this sense quasi-interpolation is asymptotically as good as least-squares approximation. We will see later on that it also compares very favorably in the non-asymptotic regime, i.e., for a given finite h.

2.1 Quasi-Interpolation in Shift-Invariant Spaces

A PSI (Principal Shift Invariant) space is a space spanned by the integer translates of one function [6, 7, 8]. We restrict our attention to the case of a bivariate function Φ which is continuous and compactly supported, e.g., the bicubic B-spline. The PSI space S_Φ is defined as the span of the integer translates of Φ,

$$S_\Phi = \operatorname{span}_{\alpha\in\mathbb{Z}^2}\{\Phi(\cdot - \alpha)\}.$$

A scale of spaces with increasing resolution is defined by

$$S_\Phi^j = \operatorname{span}_{\alpha\in\mathbb{Z}^2}\left\{\Phi(2^j\cdot - \alpha)\right\} = \left\{f(2^j\cdot) \mid f \in S_\Phi\right\},$$

where j denotes the *level* ($j = 0$ is coarsest). As j increases, the space S_Φ^j captures finer scales. For subdivision surfaces, the underlying function Φ is *refinable*, namely, it can be written as a combination of dilates and translates of itself. Therefore we have $S_\Phi^j \subset S_\Phi^{j+1}$.

We measure the approximation error in the *maximum norm*, $\|f\|_\infty = \max_x f(x)$, and denote $b^j(f) \in S_\Phi^j$ as the best approximation to f from the space S_Φ^j. It can be shown that if S_Φ contains all of the bivariate polynomials of degree less than $m+1$, then for every function f which has $m+1$ continuous and bounded derivatives, we have

$$\left\|b^j(f) - f\right\|_\infty = O\left(2^{-j(m+1)}\right), \quad j \to \infty, \qquad (2)$$

and we say that the scale of spaces S_Φ^j has approximation order $m+1$. The converse is also true: in order to get approximation order $m+1$, S_Φ has to contain all bivariate polynomials of degree less than $m+1$. In that sense Eq. (2) gives an optimal error estimate.

A *quasi-interpolation operator* Q^j is defined by a locally supported mask $(a_\beta)_{\beta\in\mathbb{Z}^2}$, which is applied to samples of the given function [4]. For $j = 0, 1, \ldots$, Q^j calculates a function in S_Φ^j that approximates f,

$$Q^j f = \sum_{\alpha\in\mathbb{Z}^2}\left(\sum_{\beta\in\mathbb{Z}^2} a_\beta f(2^{-j}(\beta+\alpha))\right)\Phi(2^j\cdot - \alpha).$$

If the mask a contains only a few non-zero elements, then Q^j computes the coefficients of each basis function using a weighted average of only a small number of samples of f.

If a quasi-interpolation operator reproduces all polynomials π up to and including degree m, i.e., $Q^0\pi = \pi$ then

$$\left\|Q^j f - f\right\|_\infty \le c2^{-j(m+1)}\|f^{(m+1)}\|_\infty \sum_\beta |a_\beta|, \qquad (3)$$

for all functions f that have $m+1$ continuous derivatives, for $j = 0, 1, \ldots$. The constant c depends only on the support of the mask a. Thus a quasi-interpolation operator can achieve the optimal approximation order $m+1$. The only condition is that Q is exact for polynomials up to the maximal degree of polynomials in the space S_Φ. Fortunately, it is not difficult to construct such operators.

Although Q^j uses only samples of the function, it can be shown that $Q^j f$ converges to f not only in the maximum or l_2 norms, but also in Sobolev norms, i.e., derivatives of $Q^j f$ converge to derivatives of f.

2.2 Quasi-Interpolants for Regular Meshes

For regular meshes, Catmull-Clark surfaces are exactly bicubic B-spline surfaces. The corresponding PSI space is spanned by the bicubic B-spline Φ with support in $[-2,2]^2$. That space contains all cubic polynomials, but not all quartics. Therefore quasi-interpolation must only be exact for cubic polynomials, $m+1 = 4$, to achieve the optimal approximation order.

Quasi-interpolation stencils which are supported on the 1-ring of a given vertex are simple to derive in the regular setting. Due to symmetry we need to choose coefficients a, b, c (see Figure 2b) so that the resulting stencil produces the proper B-spline control point values for constant ($p(x,y) = 1$) and quadratic ($p(x,y) = x^2$) functions. Reproduction of linears and cubics will follow from symmetry. These two requirements are satisfied for all simultaneous solutions of

$$\begin{aligned} 1 &= 4a + 4b + c \\ -1 &= 12a + 6b. \end{aligned}$$

1	4	1
4	16	4
1	4	1

(a)

a	b	a
b	c	b
a	b	a

(b)

-1	-4	-1
-4	56	-4
-1	-4	-1

(c)

Figure 2: *(a) The limit stencil for bicubic splines. (b) Notation for coefficients for quasi-interpolation operators. (c) The quasi-interpolation operator of our choice: $Q = 2I - L$. Note that the stencils are not normalized.*

Arbitrarily setting $a = 0$, for example, results in a stencil which requires only 5 function evaluations. The minimum l_1 norm solution, which results in the least constants in Equation (3) is achieved for $b = 0$. However, the natural generalization of the univariate setting is given by the stencil in Figure 2c. To see this, observe that in the univariate cubic B-spline setting (Eq. 1) we have weights

$$(-1/6, 8/6, -1/6) = (0, 2, 0) - (1/6, 4/6, 1/6),$$

which we may write as $Q = 2I - L$ where I is the identity and L is the limit stencil for cubic B-splines. The limit stencil for bicubic B-splines is the tensor product of the univariate limit stencil (Figure 2a). We can now see that Figure 2c is equal to $2I - L$. Even though its l_1 norm is not minimal we prefer it since it has an obvious generalization to the irregular setting (see below).

2.3 A Heuristic for Irregular Meshes

Since there is no approximation theory for subdivision schemes on irregular meshes, we tried different heuristics to extend quasi-interpolation to irregular vertices. For all the masks we tried, we found increased approximation error near the irregular vertices. This comes as no surprise since Catmull-Clark subdivision surfaces cannot even reproduce all quadratic functions at the irregular vertices. With all trials yielding similar results, we chose a heuristic that is simple to implement. Consider the quasi-interpolation operator in Figure 2c which can be written as twice the identity minus the *limit mask*

$$Q = 2I - L. \tag{4}$$

This formula carries over to the irregular setting, including common feature rules [1, 16], in a straightforward fashion. The identity (I) for a given vertex in the control mesh is the stencil with 1 at the given vertex and 0 at all other vertices. Limit stencils (L) are known for all popular rules such as creases, boundaries, etc. (see for example [1]). Together they define Q as above everywhere in a control mesh with regular control points leading to the standard stencil (Figure 2c).

To see why our quasi-interpolant in Equation (4) is a reasonable choice, it is helpful to consider the approximation problem from a different point of view. Given a control mesh that samples the target surface Ω at points $\omega_i \in \Omega \subset \mathbb{R}^3$, the usual interpolation problem is to find $\bar{p}$ such that $\bar{\omega} = L\bar{p}$. The vectors $\bar{\omega}$ and $\bar{p}$ hold the samples ω_i and the control points p_i of the control mesh, respectively, while L contains the limit operators for each of the control points. Ignoring questions of rank for the moment, the interpolation problem may be solved as $\bar{p} = L^{-1}\bar{\omega}$. In general, L^{-1} is dense, making this approach expensive. However, we can find an *approximate* solution for L^{-1} as follows. Since $L = I - (I - L)$ and $\|I - L\| < 1$, the use of a Neumann series gives

$$L^{-1} = \sum_{i=0}^{\infty}(I - L)^i = I + (I - L) + (I - L)^2 + \ldots$$

Using only the first two terms we find

$$L^{-1} \approx 2I - L.$$

Hence Q as defined in Equation (4) amounts to a "first order approximation" of the inverse of the limit operator L. This provides some theoretical insight into the good performance of our quasi-interpolant in the irregular setting that is seen in our experiments.

3 The Surface Fitting Algorithm

The input to the fitting algorithm consists of a subdivision surface s_{in} with detail coefficients and the target surface which is given as a function $\phi : s_{\text{in}} \to \mathbb{R}^3$ mapping every point on s_{in} to a new point in space. ϕ differs from the identity only in the local area of the surface that is being fitted. The user also prescribes a collection of mesh faces, A, that designate areas of the surface not to be modified. A tolerance on the distance between the target surface and the approximation must also be provided[1].

We use Catmull-Clark subdivision surfaces with detail coefficients

$$p^{j+1} = Sp^j + d^{j+1}, \quad j = 0, 1, \ldots,$$

where S is the subdivision operator, p^j are the control points of the mesh at level j, and d^{j+1} are detail coefficients at level $j + 1$. Note that this implies that detail coefficients are not critically sampled as would be the case in wavelet constructions. Instead they are more akin to what one would find in a Laplacian pyramid or MIP map. Each control point is associated with some vertex v. For a subdivision surface s and a vertex v at an arbitrary level of subdivision, $s(v)$ denotes the associated limit position. The point on the target surface corresponding to v is $\phi(s(v))$. Thus s and ϕ define a *correspondence* between every vertex of the mesh at any level, and some point on the target surface. The correspondence determines how the target surface is sampled at the vertices, and therefore directly affects the approximation error, which we control by adaptively subdividing to increase the sampling density where necessary. Figure 3 shows the correspondence for a simple curve and a uniform cubic B-spline as it is alternately fitted to the curve and subdivided.

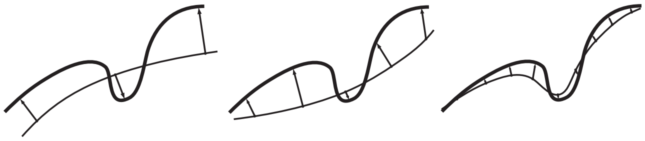

Figure 3: *The correspondence for a simple curve (thick line) and a uniform cubic B-spline (thin line) is illustrated at the knot values by the arrows. The B-spline is fitted to the curve at successively finer resolutions to improve the sampling.*

The quasi-interpolation operator (Eq. 4) can be applied to samples of the function, but we have observed smaller approximation errors when applying it to the *difference*, $\phi(s) - s$. We emphasize that the smaller approximation errors are an empirical observation and we are not aware of any theoretical justification. One nice side effect of this approach is that it ensures zero details if the surface already approximates the desired function, i.e., an interpolating solution remains unchanged under this modified approach.

3.1 Adaptive Approximation

The adaptive approximation algorithm creates a sequence of surfaces s^j for $j = 0, 1, \ldots, k$ that approach the target surface $\phi(s_{\text{in}})$ starting from s_{in} and stopping at level k when s^k satisfies the approximation requirements. All of the surfaces have the same control mesh as s_{in}. The algorithm is initialized at level zero as follows:

[1] The user could also prescribe a tolerance on the difference between the normal vectors of the target and approximating surface.

1. Let V^0 be the collection of vertices at level 0 that do not influence the limit surface at any patch in A. Let B^0 denote the complement of A.

2. Set $s^0 := s_{\text{in}} + Q^0(\phi(s_{\text{in}}) - s_{\text{in}})$, i.e., for every vertex v in V^0, $p^0(v) = p^0(v) + Q^0(\phi(s_{\text{in}}) - s_{\text{in}})(v)$.

For $j = 0, 1, \ldots$ do

1. Let C^j be the set of patches influenced by any $v \in V^j$ and let $B^{j+1} := \emptyset$.

2. For any patch in $C^j \cup B^j$ which fails the approximation criterion, add its children to B^{j+1} and stop if $B^{j+1} = \emptyset$.

3. Let V^{j+1} be the vertices at level $j+1$ that influence at least one patch in B^{j+1} but none in A.

4. Set $s^{j+1} := s^j + Q^{j+1}(\phi(s_{\text{in}}) - s^j)$, i.e., for every vertex v in V^{j+1}, $d^{j+1}(v) = Q^{j+1}(\phi(s_{\text{in}}) - s^j)(v)$.

The termination of the algorithm is controlled by the approximation criterion. However, the basic structure of the algorithm is independent of the particular design of the criterion. For our examples, we sample the error over the given patch at a discrete set of points, taken at vertices of its subpatches. The approximation is deemed satisfactory when the error is within the provided tolerance.

3.2 Comparison of Least-Squares and Quasi-Interpolation

Quasi-interpolation is optimal in the asymptotic sense in the regular setting. However, a single level of quasi-interpolation does not give a best approximation in any of the norms that are typically of interest. To test the performance of quasi-interpolation we compared it with a best approximation in the least-squares norm: given some s_{in} and a function ϕ, we calculated s_l and s_q by least-squares and quasi-interpolation respectively. The least-squares norm is approximated discretely at level 2 vertices as

$$E(s) = \|s - \phi(s_{\text{in}})\|_2 = \left(\sum_{v \in V^2} |s(v) - \phi(s_{\text{in}}(v))|^2 / n \right)^{1/2}$$

where n is the number of vertices. s_l is defined as the minimizer of E, while

$$s_q = s_{\text{in}} + Q^0 \left(\phi(s_{\text{in}}) - s_{\text{in}} \right).$$

We repeat the experiment at increasingly finer resolutions, using s_q from the previous level as s_{in} for quasi-interpolation at the next finer resolution. For s_l no initial surface is needed.

The results of our experiments are reported in Table 1 for a set of functions $z = f(x, y)$, $(x, y) \in [-1, 1]^2$, chosen to examine the behavior of the approximation under a variety of conditions: a smooth function with decay at infinity (Fig. 4a); a smooth function with oscillations (Fig. 4b); a function with a C^1 discontinuity at the origin (Fig. 4c); and a function with a C^1 discontinuity along a curve (Fig. 4d). A regular grid with 10 divisions on $[-1, 1]^2$ served as s_{in}. Exemplary pseudo-color plots of the error are shown in Fig. 5. The error near singularities decays faster for quasi-interpolation, due to the local support of the stencil. The overall rate of decay of the error for least-squares and quasi-interpolation is the same. In the two cases of approximation to non-smooth functions, quasi-interpolation yields an error which is very close to the optimal least-squares error, demonstrating that the constants in the error bound are indeed low.

We examined the behavior of the approximation in the irregular setting for the function in Fig. 4a. A surface with $k \geq 3$ regular regions meeting at the origin was used for s_{in}. The error over the domain $x^2 + y^2 \leq 1$ for the least-squares and quasi-interpolation approximations is plotted in Fig. 6 for $k = 3, \ldots, 13$. Exemplary pseudo-color plots of the error are shown in Fig. 7.

		10×10	20×20	40×40
LSQ QI	$z = e^{-9(x^2+y^2)}$	1.067e-3 9.428e-3	3.103e-5 3.164e-4	1.545e-6 1.643e-5
LSQ QI	$z = \sin(\pi x)\sin(\pi y)$	2.438e-4 7.832e-3	7.164e-6 5.507e-5	4.022e-7 5.680e-6
LSQ QI	$z = \sqrt{x^2 + y^2}$	1.209e-3 1.674e-3	3.092e-4 6.394e-4	8.927e-5 1.688e-4
LSQ QI	$z = \max(1/2 - x^2 + y^2, 0)$	7.613e-3 1.153e-2	2.620e-3 3.771e-3	9.419e-4 1.345e-3

Table 1: *Least-squares (LSQ) and quasi-interpolation (QI) errors.*

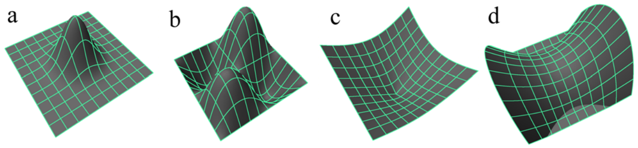

Figure 4: *(a)* $z = e^{-9(x^2+y^2)}$ *(b)* $z = \sin(\pi x)\sin(\pi y)$ *(c)* $z = \sqrt{x^2 + y^2}$ *(d)* $z = \max(1/2 - x^2 + y^2, 0)$.

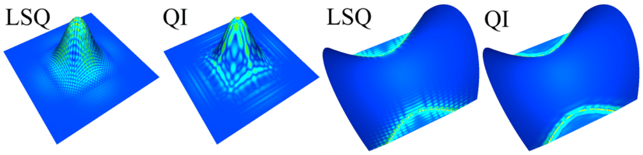

Figure 5: *Exemplary pseudo-color plots of the error for least-squares approximation (LSQ) and quasi-interpolation (QI) for functions in Fig. 4a (left) and Fig. 4d (right) at resolution 40.*

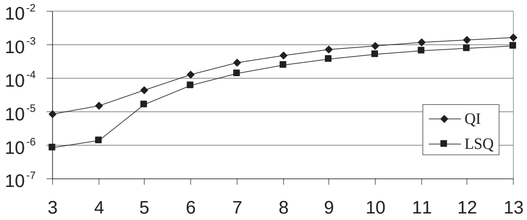

Figure 6: *The quasi-interpolation and least-squares error plotted on a logarithmic scale for irregular surfaces* $k = 3, \ldots, 13$.

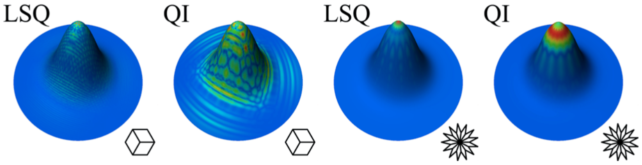

Figure 7: *Exemplary pseudo-color plots of the error for least-squares approximation (LSQ) and quasi-interpolation (QI) of the function in Fig. 4a for irregular surfaces* $k = 3$ *and* $k = 13$.

The error near the irregular vertex grows as k increases. The error at the irregular vertex scales as $O(h^2)$ since the surface does not reproduce all quadratics there. However, the error is dominated by the behavior away from the irregular vertex, where the surface is regular. For this reason, the error increases less rapidly for the quasi-interpolation approximation due to the local support of the stencil. This is most apparent for large values of k.

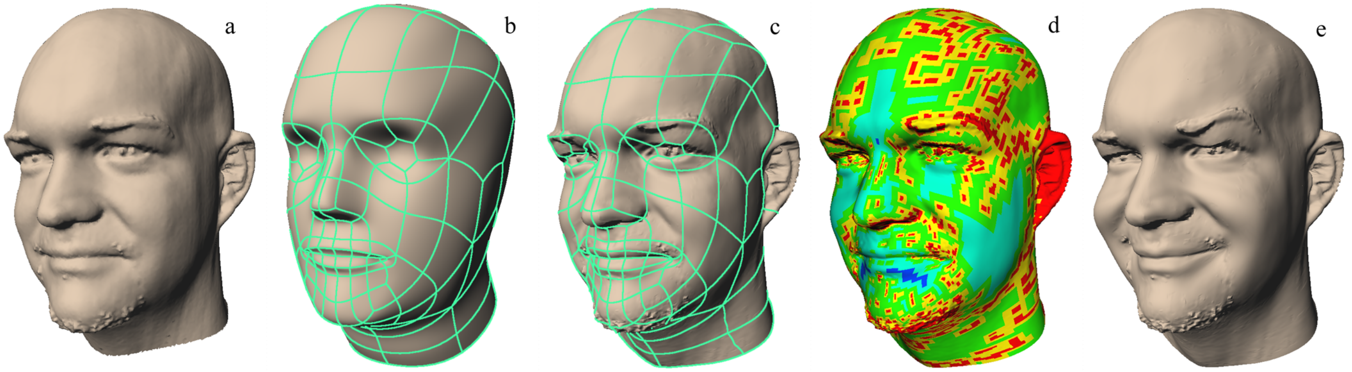

Figure 8: *(a) An original Cyberware scan with 300k triangles. (b) A simple subdivision surface for a generic head. (c) Fit of the head in (b) to the original data in (a). (d) Illustration of the depth of non-zero detail coefficients (hotter signifies finer scales). (e) Result of editing the coarsest patch level (b). To achieve correct editing semantics all details were expressed in local coordinate frames* during the edit. *Otherwise all detail computation are always performed in a world space coordinate frame.*

4 Examples

We have applied our method of fitting subdivision surfaces to surface modeling, data fitting, and scientific visualization. Statistics for the featured models are given in Table 2. All calculations were performed on an 600Mhz Intel Pentium III.

Model	Base Polygons	Base Vertices	Max. Depth	Details
fluidsim	98	109	3	8484
head	140	153	6	109022
jug	43	46	4	2060
pad	98	104	8	121362
perfume	6	8	8	31125

Table 2: *Statistics for the fitting examples.*

Cross-Section Editing Surfaces may be sculpted by modeling with cross-sectional curves (Fig. 9). An initial curve is computed by taking a cross-sectional slice of the model. The user then directly manipulates the surface by shaping the curve to the desired profile. The surface is deformed in a local region around the cross-section, by blending with the profile curve using a C^2 blend function that goes to zero at a user-specified distance from the cross-section. At the cross-section the profile curve is interpolated to high precision. All manipulations were performed interactively.

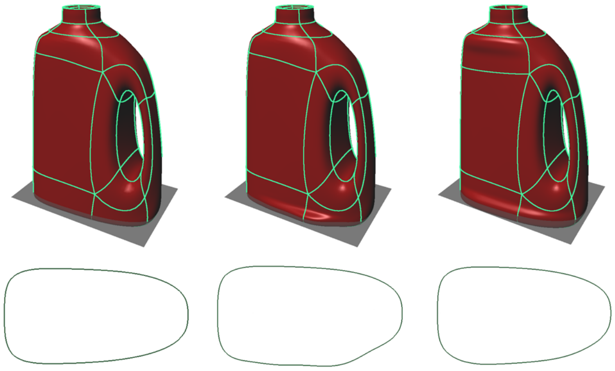

Figure 9: *Surface modeling with cross-section curves, illustrated below each model.*

Embossing Fine detail on the surface can be modeled in relief with an embossing stencil (Fig. 10). The stencil is provided by a bitmap that defines the height of the emboss. The surface is deformed by projecting the stencil on the surface and quasi-interpolating the associated height profile. The inset in Fig. 10 illustrates how the approximation adapts to features of varying detail. This process is performed adaptively both on the surface and in the bitmap to ensure that excessive refinement is not necessary. Contrary to visual techniques such as bump mapping, we provide a unified model for both fine and coarse geometric detail, suitable for manufacturing purposes.

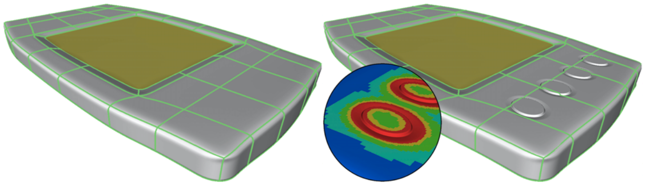

Figure 10: *Fine surface detail is added by embossing. The inset illustrates the local adaptation to varying detail (hotter signifies finer scales).*

Data Fitting Laser scanning allows very detailed objects to be captured, but the data is typically far too complex to edit directly. By fitting a simple subdivision surface to the scanned data, the object can be manipulated much more intuitively while retaining its intricate detail. We demonstrate this approach by fitting a subdivision surface to laser scan data given as a dense triangle mesh (300k triangles; Fig. 8a). The artist first constructs an initial subdivision surface with a basic patch layout suitable for later animation (Fig. 8b). Since the data was acquired by a cylindrical scan, we used simple cylindrical coordinates for the correspondence[2]. After alignment, the system computes the detail coefficients to produce the subdivision surface in Fig. 8c. The fit is adaptive in areas with finer detail ("hotter" colors in Fig. 8d). Now the artist can manipulate the fully detailed model using the control points from the original surface (Fig. 8e). The fit was computed in 6 seconds, to a relative tolerance of 0.03%.

Scientific Visualization The rapid visualization of an evolving physical system is helpful in simulation settings. Due to the fine resolution of high fidelity simulations, I/O bandwidth off of highly parallel machines is often a very constrained resource. Saving every frame of a level set simulation, for example, may not even be possible. Subsequent processing of the time-varying volume data using traditional methods, such as marching cubes iso-surface extraction followed by decimation of the meshes, to adapt to available rendering resources is awkward and expensive, both in time and storage. An alternative approach can be based on coarsely sampling an iso-surface of interest, followed by hierarchical refinements where desired. Using an appropriate correspondence to sample the data, we

[2] We switch to polar coordinates to resolve the singularity at the pole.

construct a smooth surface at successively finer resolutions, subject to time and memory restrictions. Figure 11 demonstrates this idea in the case of a level set simulation of two fluids of differing density mixing under gravity, shown by the interface between the fluids [3]. At the coarsest level, a base mesh is constructed from 109 sample points. A naïve algorithm that constructs a mesh from the sample points themselves is demonstrated in Fig. 11a. Contrast this to the surface which we can generate based on the same sample points using quasi-interpolation (Fig. 11b). The surface is fit at finer resolutions as more sample points are taken in Fig. 11c-d. The final surface in Fig. 11d was generated in 300ms. This example illustrates how the same data can be used to reconstruct the surface of interest with a much higher quality using quasi-interpolation rather than piecewise linear interpolation. This is especially evident at low numbers of samples.

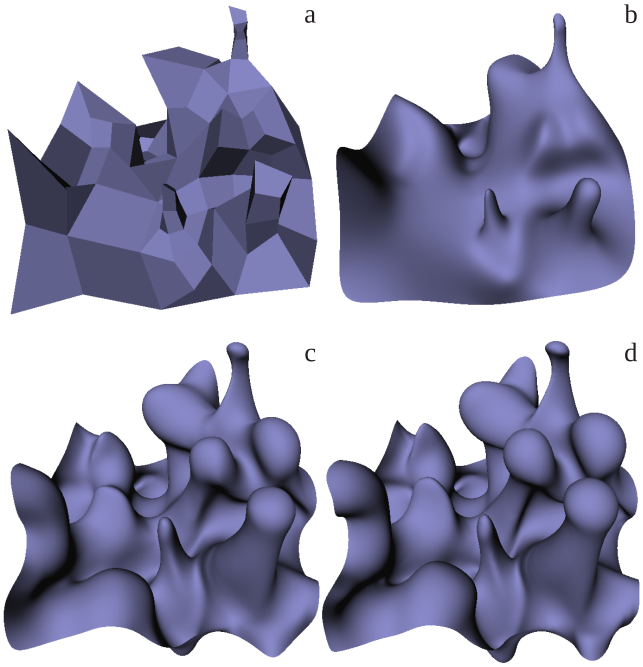

Figure 11: *Visualization of a fluid simulation. a) A mesh constructed from the initial 109 sample points. b) Our surface, generated by fitting to the same sample points. Additional detail is added to the surface after c) 1609 samples and d) 6353 samples.*

5 Contributions and Outlook

We have presented an extension of quasi-interpolation (QI) to Catmull-Clark surfaces with details, and used it as the basis for a simple and effective approximation algorithm. Due to the locality of QI, the proposed approximation scheme can be used to apply local deformations to a surface at low computational cost and with a guaranteed error tolerance. To demonstrate the power of these operators we have given some examples of approximation based on them.

Since QI does not involve the solution of linear systems, it is significantly simpler and safer to use in comparison to other common techniques, such as interpolation and least-squares. We reviewed theoretical results about the optimality of QI, and added empirical evidence showing that QI behaves very well in comparison to the optimal least-squares method.

The appeal of our algorithm is its ability to generate a suitable approximation of any shape that possesses an appropriate sampling method, as demonstrated by our examples. Locality and the coarse-to-fine fitting strategy enable many applications of this technique in scientific simulation and visualization, which promise to be much more efficient than traditional approaches (e.g., marching cubes extraction followed by decimation).

In the area of geometric modeling our techniques allows the efficient creation of highly detailed features such as embosses or the editing of cross-section curves without regard to alignment of patch boundaries or iso-parameter lines. In this way QI offers an opportunity to abstract from the underlying representation during editing. This should prove beneficial to geometric modeling user interfaces which aim to abstract from the underlying representation.

In future work we hope to develop a theory for optimal approximation in the irregular setting and explore other applications of QI in digital geometry processing.

Acknowledgment This work has been supported in part by NSF (DMS-9874082, DMS-9872890, ACI-9982273), Alias|Wavefront, Pixar, Microsoft, Intel, Lucent, and the Packard Foundation. Special thanks to Geoff Banner, Khrysaundt Koenig, Rick Kogucki, and Cory Mogk for modeling, lighting, and texturing, and to Igor Guskov and Santiago V. Lombeyda for help with the fluids dataset. Datasets are courtesy of Cyberware as well as Andy Cook and Paul Dimotakis.

References

[1] Biermann, H., Levin, A., and Zorin, D. Piecewise Smooth Subdivision Surfaces with Normal Control. *Proceedings of SIGGRAPH 2000* (2000), 113–120.

[2] Conrad, B., and Mann, S. Better Pasting via Quasi-Interpolation. In *Curve and Surface Design: Saint-Malo 1999* (1999), pp. 27–36.

[3] Cook, A. W., and Dimotakis, P. E. Transition Stages of Rayleigh-Taylor Instability Between Miscible Fluids. *J. Fluid Mech.* (to appear).

[4] deBoor, C. Quasiinterpolants and Approximation Power of Multivariate Splines. In *Computation of Curves and Surfaces*, M. Gasca and C. A. Michelli, Eds. Kluwer Academic Publishers, 1990, pp. 313–345.

[5] deBoor, C., DeVore, R. A., and Ron, A. On the Construction of (Pre)Wavelets. *Constructive Approximation 9* (1993), 123–166.

[6] deBoor, C., DeVore, R. A., and Ron, A. Approximation from Shift-invariant Subspaces of $L_2(\mathbb{R}^d)$. *Trans. Amer. Math. Soc. 341* (1994), 787–806.

[7] deBoor, C., DeVore, R. A., and Ron, A. The Structure of Finitely Generated Shift-invariant Subspaces of $L_2(\mathbb{R}^d)$. *J. Functional Anal. 119* (1994), 37–78.

[8] deBoor, C., DeVore, R. A., and Ron, A. Approximation Orders of FSI Spaces in $L_2(\mathbb{R}^d)$. *Constructive Approximation 14* (1998), 411–427.

[9] DeRose, T., Kass, M., and Truong, T. Subdivision Surfaces in Character Animation. *Proceedings of SIGGRAPH 98* (1998), 85–94.

[10] Halstead, M., Kass, M., and DeRose, T. Efficient, Fair Interpolation Using Catmull-Clark Surfaces. *Proceedings of SIGGRAPH 93* (1993), 35–44.

[11] Khodakovsky, A., and Schröder, P. Fine Level Feature Editing for Subdivision Surfaces. *ACM Solid Modeling Symposium* (1999), 203–211.

[12] Lee, A., Moreton, H., and Hoppe, H. Displaced Subdivision Surfaces. *Proceedings of SIGGRAPH 2000* (2000), 85–94.

[13] Lee, A. W. F., Sweldens, W., Schröder, P., Cowsar, L., and Dobkin, D. MAPS: Multiresolution Adaptive Parameterization of Surfaces. *Proceedings of SIGGRAPH 98* (1998), 95–104.

[14] Levin, A. *Combined Subdivision Schemes*. PhD thesis, Tel-Aviv University, 2000.

[15] Litke, N., Levin, A., and Schröder, P. Trimming for Subdivision Surfaces. *Computer Aided Geometric Design 18*, 5 (June 2001), 463–481.

[16] Schweitzer, J. E. *Analysis and Application of Subdivision Surfaces*. PhD thesis, University of Washington, 1996.

[17] Zorin, D., and Schröder, P., Eds. *Subdivision for Modeling and Animation*. Course Notes. ACM SIGGRAPH, 2000.

[18] Zorin, D., Schröder, P., and Sweldens, W. Interactive Multiresolution Mesh Editing. *Proceedings of SIGGRAPH 97* (1997), 259–268.