

Smooth adaptive fitting of 3D models using hierarchical triangular splines

Alex Yvart, Stefanie Hahmann, Georges-Pierre Bonneau

▶ **To cite this version:**

Alex Yvart, Stefanie Hahmann, Georges-Pierre Bonneau. Smooth adaptive fitting of 3D models using hierarchical triangular splines. International Conference on Shape Modeling and Applications, SMI'05, Jun 2005, Boston, United States. pp.13-22. hal-00319669

HAL Id: hal-00319669
https://hal.science/hal-00319669

Submitted on 9 Sep 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Smooth Adaptive Fitting of 3D models using hierarchical triangular splines

Alex Yvart
LMC-IMAG
Grenoble, France
Alex.Yvart@imag.fr

Stefanie Hahmann
LMC-IMAG
Grenoble, France
Stefanie.Hahmann@imag.fr

Georges-Pierre Bonneau
GRAVIR-IMAG
Grenoble, France
Georges-Pierre.Bonneau@imag.fr

Abstract

The recent ability to measure quickly and inexpensively dense sets of points on physical objects has deeply influenced the way engineers represent shapes in CAD systems, animation software or in the game industry. Many researchers advocated to completely bypass smooth surface representations, and to stick to a dense mesh model throughout the design process. Yet smooth analytic representations are still required in standard CAD systems and animation software, for reasons of compactness, control, appearance and manufacturability. In this paper we present a new method for fitting a smooth adaptively refinable triangular spline surface of arbitrary topology to an arbitrary dense triangular mesh. The key ingredient in our solution is that adaptive fitting is achieved by 4-splitting triangular surface patches locally therefore no particular attention has to be paid the validity of an underlying subdivided mesh. Furthermore, the final surface is composed of low-degree polynomial patches that always join with G1-continuity. The ability to adaptively refine the model allows to achieve a given approximation error with a minimal number of patches.

1. Introduction

3D laser scanning systems are capable of producing detailed and densely sampled triangular meshes. On one hand triangle meshes are an adequate representation for many applications, for example in entertainment industries with computer animations for movies and publicity, or display of rapidly changing scenes in video games. Recent advances in mesh processing algorithms including editing, simplification, denoising, compression have made possible the use of large and highly detailed models.

On the other hand smooth surface representations may be required for reasons of compactness, control, appearance and manufacturability. The creation of a complex object of arbitrary topology with a lot of details is still a laborious process despite the use of advanced Computer Aided Design (CAD) and Computer Graphics modeling systems.

3D scanning of existing objects or clay models is thus offering an utile alternative for model acquisition. The whole process of converting point clouds as output of 3D scanners into a geometric model is called surface reconstruction. It is generally preceded (or combined with) a registration step where range images are merged into a consistent large set of unorganized points. The geometric model can either be a polygonal mesh [3, 38, 19, 37, 6, 1], a physical based model [35, 24, 9], or a smooth surface. Reconstruction of a smooth surface is often referred to as *surface fitting*, i.e. interpolation or approximation of unorganized points or organized (gridded or triangulated) data. The resulting surface can be represented as NURBS [30, 31, 33], or implicit surfaces [32, 28, 5, 36]. The problem of fitting (or reconstructing) smooth surfaces of arbitrary topological type has been addressed in some recent work using surface splines [11], B-splines combined with displacement maps [21], subdivision surfaces [18, 25], algebraic surfaces [27, 2], and hierarchical B-splines [13], see also Section 2.

In this paper we present a method for fitting a smooth adaptively refinable triangular spline surface of arbitrary topology to an arbitrary dense triangle mesh.

The main features and contributions of this method are:

Surface representation. The surface is represented explicitly in piecewise polynomial or spline form. It consists of triangular patches, each of them parameterized over the unit triangle. An advantage of triangle based methods is that most available dense meshes are triangle meshes. No conversion of triangle meshes into quadrilateral meshes is necessary.

Adaptive fitting. The fitting process is adaptive and capable to satisfy a user specified error tolerance. A coarse base mesh is interpolated by an initial G^1-continuous surface. This initial surface is then adaptively refined. The affine image of the base mesh serves as parameter domain for the initial surface and for all subsequent refined surfaces. In order to locally refine the surface, pairs of adjacent triangles in the base mesh are 4-splitted, correpondingly new

smooth patches are computed. Note that this adaptive refinement procedure does not require a valid triangulation as parameter domain (i.e. a 2-complex manifold). In particular there may be T-junctions in the parameter domain and there may be coarse smooth patches being adjacent to several finer smooth patches, as illustrated in Figure 1. This distinguishes our work from previous papers that perform adaptive fitting by successively refining a coarse mesh such that it always remains a valid mesh followed by global or local re-fitting.

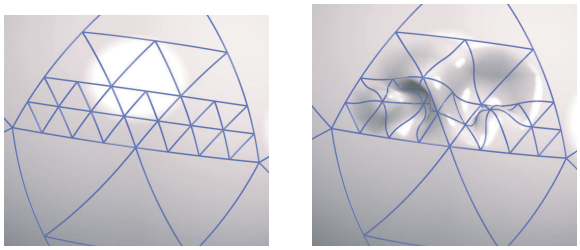


Figure 1. Left: Locally refined G^1 continuous surface. T-junctions of the parameter domain may occur. Thus coarse patches may be adjacent to several finer patches. Right: Degrees of freedom of the refined patches serve to fit finer details.

G^1 continuity. The resulting surface is overall G^1 continuous. The adaptive refinement process ensures automatically that the new refined surface patches join with G^1 continuity the remaining surface patches around. As a consequence, fitting the surface to the data is performed by a sparse least squares minimization without any constraint.

Correspondence problem. The correspondence problem is solved in several steps. First a base mesh is constructed via topology preserving mesh simplification. It serves as parametric domain. The fine mesh is then partitioned by computing boundary curves corresponding to the edges of the base mesh. We use a Dijkstra algorithm to find an initial triangle strip together with a curve smoothing algorithm [4] in order to stretch the resulting curves. Each triangular domain of the fine mesh is then parameterized over the base mesh following Desbrun et al. [8] and using [10] for smoothing.

Arbitrary topology. The method works for manifold meshes of arbitrary topological type.

2. Related Work

Fitting smooth surface of arbitrary topology has been proposed for different surface representations. We will discuss them in this Section by emphasizing the difference to our method. Furthermore let us state that technical smoothness in form of G^1-continuity as well as visual pleasant smoothness is an essential requirement for obtaining high quality and manufacturable surfaces. Our fitting method has to be placed into that context.

Most recent work is related to three types of smooth surfaces, all capable to deal with arbitrary topologies: subdivision surfaces, algebraic surfaces and parametric piecewise polynomial surfaces.

Hoppe et al. [18] adapted Loop's subdivision scheme to fit an unorganized set of points by a piecewise smooth surface, thus able to model sharp features. Later Lee et al. [22] used subdivision surfaces with displacement maps. Halstead et al. [16] described interpolation with Catmull-Clark surfaces. Fitting Catmull-Clark surfaces using quasi-interpolation has been introduced by Lee et al. [25]. All methods support surfaces of arbitrary topological type by nature. They work either with arbitrary quad meshes or with triangle meshes. Whereas subdivision surfaces are well suited for animation and display purposes, they are not commonly supported within current CAD modeling systems.

Fitting G^1 piecewise algebraic surfaces of arbitrary topology from an unorganized set of points was described by Bajaj et al. [2] and Moore and Warren [27]. A surface patch is obtained by fitting an algebraic patch to the data within a tetrahedron. The methods are adaptive. The resulting surface is piecewise implicit.

The most popular smooth surface representations are tensor product NURBS. Many previous works exist for gridded data as well as for irregular data. let us focus only on techniques that are related to fitting G^1 surfaces of arbitrary topological type. Eck and Hoppe [11] described a method to fit irregular meshes (constructed from unorganized data using [19, 20] with automatically placed bicubic Bézier patches. G^1 continuity is achieved by using Peters' surface splines [29] which are a suitable representation to model surface of arbitrary topology. An extra step is necessary to transform a triangular mesh into a quad mesh. The method is adaptive in the sense that any quad mesh can be locally subdivided.

Our method in contrast works directly with triangle meshes, since the surface is defined by piecewise polynomial triangular Bézier patches. By nature triangular patches are particularly well adapted to represent arbitrary topologies. Furthermore we don't require the mesh to be valid after local refinement (see Section 3.5).

Krishnamurthy and Levoy [21] fit B-spline surfaces of arbitrary topological type. The method uses displacement maps for capturing fine details. Thus it is more intended to display purposes. Furthermore it needs user interaction to delineate the patch boundaries. Very little discussion is given on how to join the B-spline patches with G^1-continuity in a non-tensor product configuration. Stitching

together B-spline patches using geometric continuity conditions [12] is not a trivial problem, in particular at vertices of valence $\neq 4$. Shi et al. [34] bypass that problem by using approximate G^1 conditions.
Our approach guarantees automatically a G^1 surface construction through all levels of local refinement. No extra linear or non-linear constraints need to be fulfilled during the adaptive fitting process.

3. Surface Fitting

The input of the fitting algorithm consists of a dense irregular triangular mesh $M_D = (K_D, V)$ of arbitrary topology (D stands for dense). K_D is a simplical complex determining the topological type of the mesh. It specifies the connectivity of the vertices, edges and faces of the mesh. $V = \{v_1, \ldots, v_{n_D}\} \subset \boldsymbol{R}^3$ are the vertex positions determining the shape of the mesh in space. The output is a triangular smooth G^1 continuous piecewise polynomial surface S of degree 5 fitting the fine mesh. The fitting pipeline consists of 5 successive steps which will be described in detail in this Section.

When fitting a parametric surface the correspondence problem has to be solved. It consists in parameterizing the dense polygonal mesh over a parameter domain by associating a domain point to each data point. For surfaces of arbitrary topology several strategies are possible. Eck et al. [10] propose a fully automatic one-stage process. A coarse domain mesh together with an initial parameterization is constructed using Voronoï tiles obtained by region growing and the use of harmonic maps. A multiresolution parameterization of polygon meshes is described by Lee et al. [23] thus constructing a hierarchy of meshes and a parameterization via mesh simplification. Krishnamurthy and Levoy [21] first compute boundary curves delineating surface regions, and then parameterize these regions over a gridded domain. This method is not automatic; it requires the user to paint manually these boundary curves. In order to make profit from some recent advances in mesh parameterization [8] we proceed in two stages. First we compute a coarse mesh which serves as the parameter domain (Sect. 3.1) then we map the edges of the coarse mesh and smooth them on the coarse mesh (Sect. 3.2). A further advantage is that the process is automatic but allows for user interaction if it's desired.

The fitting process itself is adaptive. It starts by computing an initial fitting spline surface over the coarse mesh using [15]. A piecewise triangular G^1 continuous polynomial surface is thus constructed that interpolates the vertices of the coarse mesh. To each triangular region of the fine mesh corresponds a triangular patch interpolating the three corner vertices, and approximating the interior by means of least squares. Details are then added locally by refining the smooth surface iteratively and by recomputing the degrees of freedom. This adaptive refinement and approximation process stops when the data is fitted up to a user-defined precision. The surface property of being locally refinable while always guaranteeing G^1-continuity everywhere makes it particularly appropriate to be used for adaptively surface fitting of models with many details.

3.1. Compute coarse mesh M_C

We need a coarse mesh, also called *base mesh*, for two reasons. First, M_C will serve as parametric domain for the dense mesh M_D. Second, M_C serves to construct the initial smooth surface S_0 that will then be successively refined. $M_C = (K_C, V_C)$, where $V_C \subset V$ and K_C is of same topological type as K.

In order to construct the coarse mesh, we make use of a mesh decimation algorithm based on edge collapse. For each edge we estimate an error induced by its collapse following [14]. Other cost (error) functions can be found in [20, 17]. A priority queue implemented as a heap is used to select iteratively the candidates for an edge collapse with lowest cost. After each mesh transformation by edge collapse the priority of edges in the neighbourhood are updated. Any other mesh simplification technique can be used (suppression of vertices or faces) but it is important to preserve the topological type of the dense mesh M_D. The mesh decimation stops when a user-defined number of faces is reached. The right number of faces depends mainly on the geometric complexity of the model. The trade-off we observed with our fitting algorithm is the following. Remember that the coarse mesh M_C will be interpolated by a initial smooth surface. That surface will then be refined locally in order to reproduce all fine details of the input data. If M_C is too fine, the initial smooth surface S_0 already fits well and no local refinements are necessary. This means that there would probably be some regions where less triangles would be sufficient for the same precision. If in the opposite the coarse mesh is too coarse, too much global refinements of the smooth surface would be necessary, thus slowing down the convergence rate of the fitting process. For the examples shown in Section 4 we took as coarse mesh for the bunny one with 110 faces, and one with 70 faces for the Max Planck head, see Figures 7 and 8.

3.2. Compute boundary curves

In order to parameterize the input data one needs to partition the dense mesh into regions corresponding to the triangular faces of the parameter domain (the coarse mesh). When fitting these data by a smooth surface each region is initially approximated by a triangular polynomial patch. To delineate these triangular regions a network of bound-

ary curves is computed. These curves lie on the dense mesh, thus they are polygonal curves. Since the coarse mesh vertices are a subset of the dense mesh vertices, the end points of each boundary curve are already known. The curves are then computed to link these extremities by carrying out the following two steps:

• First rough boundary curves are computed as shortest paths in a weighted graph. This graph is defined by the input mesh. Its nodes are the mesh faces, its unoriented weighted edges correspond to the edges in the mesh that have two neighbouring faces. The weight corresponds to the sum of distances from the barycenter of each face to the mid point of the common edge. The result of Dijkstra is therefore a strip of triangles from which we derive a rough boundary curve by joining the edge mid-points of these faces, see Figure 2(b).

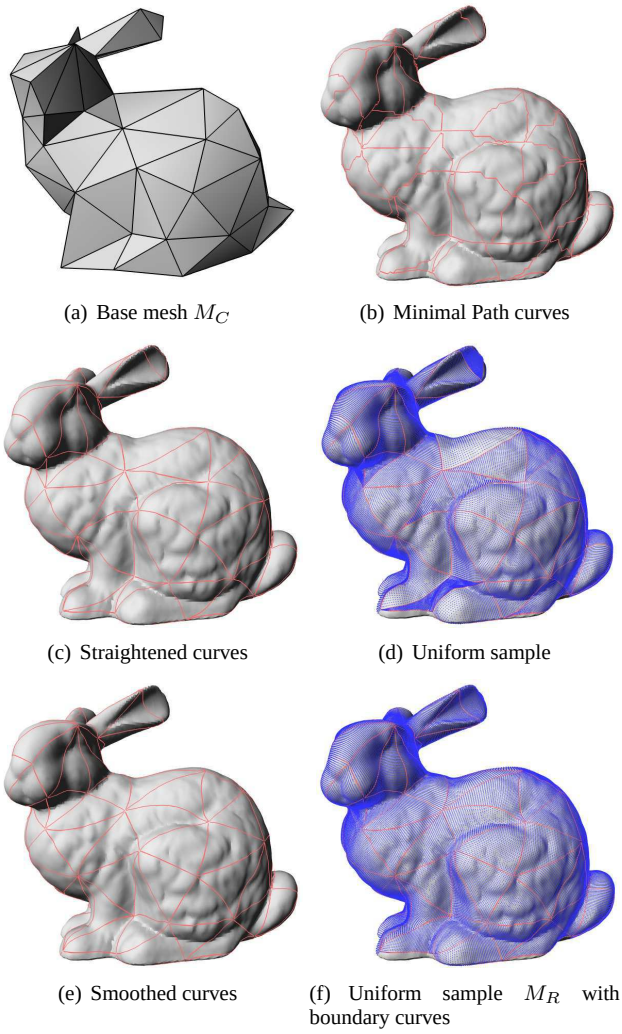
• The second step consists in straightening these rough curves iteratively [4], see Figure 2(c).

3.3. Compute parameterization

First the boundary curves computed in the previous section are parameterized over the edges of the coarse mesh using chordal parameterization. Then an **initial global parameterization** is obtained by parameterizing each triangular region of the dense mesh over a unit triangle with discrete conformal mapping following Desbrun et al. [8]. These individual region parameterizations match the boundary curve's parameterization. To each vertex of the dense mesh corresponds now a pair of parameters, i.e. by affine transformation of the parameter domain.

A **uniform sampling** is then achieved by computing the pre-image of a set of uniformly distributed points in the parameter domain. Note that this step is not absolutely necessary since the fitting can be performed as well on the non-uniform sample resulting from the initial parameterization. However this step is important in order to speed up the fitting computations. The reason is that the fitting involves a least squares approximation of type $\|Ax - b\|^2$, where x is the vector of unknown spline control points, b is the vector containing the input data. The matrix A depends on the parameterization. In the case of non-uniform sampling the matrix has to be recomputed for each patch, while with uniform sampling A needs to be computed only once.

However, parameterizing each region individually leads to distortions of the iso-parametric lines across the boundary curves, see Figure 2(d). We therefore **improve** the global quality of the parameterization by first smoothing the initial parameterization iteratively across all boundary curves similar to [10] and then sampling it uniformly. In fact a new boundary curve is computed by parameterizing two adjacent regions onto the unit square and by taking the image of the square diagonal as new boundary curve. This



(a) Base mesh M_C (b) Minimal Path curves

(c) Straightened curves (d) Uniform sample

(e) Smoothed curves (f) Uniform sample M_R with boundary curves

Figure 2. Parameterization of the Stanford bunny.

smoothing step is applied to all boundary curves in an arbitrary order. The resulting uniform resampled dense mesh is denoted $M_R = (K_R, P)$, where $P = \{p_1, \ldots, p_{n_R}\}$ are the mesh vertices, and K_R of same topological type as K_D.

3.4. Fitting the initial surface S_0

Once the correspondence problem is solved an initial smooth surface S_0 is constructed over the coarse mesh M_C fitting the input data. S_0 is composed of triangular patches which are computed in one-to-one correspondence to the faces of M_C and which interpolate the vertices V_C. At this stage one can imagine to use any method of smooth interpolation of triangulated data of arbitrary topological type using parametric surface patches, see Lounsbery et al.

[26]. However, all these schemes are designed to interpolate sparse data. They are not locally refinable and they don't have sufficient degrees of freedom in order to fit dense data without increasing unreasonably the number of patches.

We use instead the previous work of Hahmann and Bonneau [15] for fitting the initial smooth surface S_0. This method interpolates the coarse mesh M_C with G^1 continuity. It is based on a one-to-four split of the domain triangles thus producing four quintic Bézier patches in correspondence to each face of M_C. Hence several degrees of freedom allow to fit dense data with a minimum of patches. Furthermore it is local. But the main property that is distinguishing it from all previous Clough-Tocher split methods described in [26] is *local refinement*. Once the initial fitting surface computed over M_C details can be added by local refinement and local fitting.

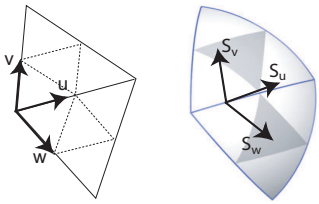


Figure 3. Left: parameter domain; right: macro-patches with tangent directions.

3.4.1. Review of the triangular spline method Let us briefly recall the main issues of that scheme by emphasizing on the free parameters. All computational details of that scheme can be found in [15]. From a coarse mesh M_C a smooth piecewise triangular spline is constructed interpolating the mesh vertices V_C. It is parameterized over the coarse mesh. Each triangle is mapped to a group of four Bézier patches of degree five. This group is referred to as a *macro-patch* Referring to the notations of Figure 3, the tangent plane continuity constraint between the two macro patches is written:

$$\Phi(u)\frac{\partial S}{\partial u} = \mu(u)\frac{\partial S}{\partial v} + \nu(u)\frac{\partial S}{\partial w}\ . \qquad (1)$$

Equation (1) states that the three partial derivatives along and across the common curve between two macro patches are coplanar, thus defining a continuous tangent plane. Φ, μ and ν are scalar functions. They are chosen to be linear, that is the lowest possible degree. (1) has to be fulfilled along all edges of the input mesh. At a mesh vertex this leads to the so-called vertex consistency problem, which states in essence that the derivatives of the surface must be consistently chosen [7]. The control points of a macro patch are illustrated in Figure 4. Different color coding correspond to different steps in the algorithm. The surface is built in four consecutive steps. We briefly elaborate these four steps focusing on the salient features, in particular the description of the free parameters.

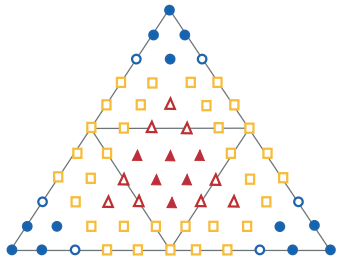


Figure 4. Control points of a macro-patch which is composed of four quintic Bézier triangles. The control points symbolized by circles are dealt with by step *(S1)*. Squared control points are computed by steps *(S2)* and *(S3)*. Step *(S4)* involves the control points symbolized by triangles.

(S1) Vertex consistency. For a vertex of order N, the free parameters are the N **first derivatives** at the vertex in the direction of each edge, and the N **twists** (mixed second partial derivatives) of the patches joining at that vertex. From these free parameters, the remaining second derivatives along the edges are computed, and the scalar functions Φ, μ and ν are fixed. Changing these parameters affects the N macro-patches joining at the vertex. In Figure 4, the free parameters correspond to the control-points symbolized by solid discs, while the circles symbolize the control-points corresponding to the second derivatives.

(S2) Boundary curves. The boundary curves are therefore piecewise quintic curves, this is the lowest possible degree fulfilling the continuity constraints. It turns out that once the derivatives at the vertices are consistently chosen by (S1) the boundary curves are completely determined, hence no free parameters in step (S2).

(S3) Cross derivatives. In order to ensure the tangent plane continuity between the macro-patches, the cross partial derivatives $\frac{\partial S}{\partial v}$ and $\frac{\partial S}{\partial w}$ have to be computed. In terms of the Bézier patches, these amounts to compute the first inner row of Bézier control points on each side of the boundary curves. There are no free parameters in step (S3). In Figure 4, the control-points computed by step (S2) and (S3) are shown as squares.

(S4) Continuity inside a macro-patch. In this step, for each macro-patch, there remain 15 unknown Bézier control points, symbolized as triangles in Figure 4. Six of these

free parameters	influence
N first derivatives and N twists at each interpolated vertex of degree N	affect all macro-patches around that vertex
6 control points inside each macro-patch	affect a single macro-patch

Table 1. The free parameters controlling the interpolating surface, and their influence on the surface. These parameters are the unknowns in the fitting process.

points can be freely chosen (the solid triangles), the 9 others are computed such as to ensure C^1-continuity between the 4 Bézier patches of the macro-patch. Changing one of these 6 free parameters affects a single macro-patch. Table 1 summarizes the free parameters and their region of influence on the surface.

3.4.2. Fitting S_0 A coarse mesh M_C and a uniform sample M_R of target points $\{p_i\}$ of the dataset have been computed. Based on this mesh, an initial surface S_0 is now constructed with the previously exposed method. Since several free parameters (see Table 1) are available, their optimal values for fitting the samples are found such that the sum of the squared distances of S_0 to the target points p_i are minimized. Since the boundary curves of the macro patches of S_0 depend non-linearly on some of the free parameters (see [15], p. 102) a global minimization would lead to a time-consuming non-linear optimization. Instead we split the optimization into two linear problems which are successively solved.

(a) the first derivatives of the patch boundary curves are estimated by computing quintic curves $c(t)$ approximating the polygonal boundary curves of M_R (see Sect. 3.2, 3.3, and Fig. 3(f)). Remember that these curves delineate the triangular regions of the re-sampled input data M_R that have to be approximated by the macro-patches. We will call the vertices of these regions *target points*. The unknown control points of $c(t)$ are obtained by solving

$$\min \sum_i \lambda_i \|p_i - c(t_i)\|^2 \qquad (2)$$

where p_i denote the target point lying on the corresponding boundary curve of M_R with parameter value t_i. The weights $\lambda_i > 0$ are chosen to be more important near the end points, since the middle of these curves will be met exactly later on when locally refining the surface. It is clear that these curves can't be taken as boundary curves of S_0 since they don't satisfy the G^1 conditions, but they provide good estimates for the first derivatives (free parameters) of the boundary curve of S_0 at the patch vertices. Furthermore, the curves $c(t)$ have been computed individually, there is no reason for the derivatives at a common vertex to be coplanar. Thus the projection onto a mean plane is finally taken as optimal input for S_0.

(b) Since the surface S_0 depends linearly on the free twists and inner control points (table 1) and since S_0 is G^1 continuous by construction, the optimal values are thus obtained by a simple unconstrained least squares fitting minimizing

$$E = \lambda E_{dist} + (1-\lambda) E_{fair}, \quad \lambda \in [0,1], \qquad (3)$$

where E_{fair} is a linearized fairness functional, and

$$E_{dist} = \sum_i \|p_i - S_0(u_i, v_i)\|^2.$$

p_i denote the target points interior to all triangualr regions of M_R wit (u_i, v_i) as parameter value.

The unknowns of (3) are the free control points of the macro patches of S_0 corresponding to the free parameters (3 twists and 6 inner control points).

3.5. Adding details

The initial fitting surface S_0 interpolating the coarse mesh does generally not present all the details of the sample. The precision of the fitting surface can be measured by computing the maximum error (distance) between the sample points and the surface patch. To reduce the error one needs to increase the number of degrees of freedom by increasing the number of surface patches. One can either globally subdivide the coarse mesh M_C, fit again and re-iterate. But this would globally increase the number of patches even where the error tolerance is already reached. Or better, one can adaptively refine the mesh locally where the error is high, fit again and re-iterate, see Figures 7(c), (f) and (i). In several previous works, like [11, 2], adaptive fitting has been performed by subdividing locally the coarse mesh followed by some local re-fitting. But the sudivided mesh must remain a valid mesh (i.e. a 2-complex manifold).

We instead introduce an adaptive fitting scheme which is based on local surface refinements. The surface is refined locally by 4-splitting a pair of adjacent triangles in the coarse mesh, and correspondingly compute new surface patches. These new surface patches join with G^1-continuity the remaining coarser surface patches. The additional degrees of freedom are used to better fit the input data. Note that in the present approach no particular attention has to be paid to the validity of an underlying mesh. T-junctions may occur in the parameter domain, so that coarse surface patches are adjacent to several finer patches along a common boundary curve, see Figure 1.

The main reason for using [15] as basic surface representation (Section 3.4) is that this triangular interpolant allows for such local surface refinement: It has been shown

in [39] that one local surface refinement is obtained by performing the following steps, illustrated in Figure 5:

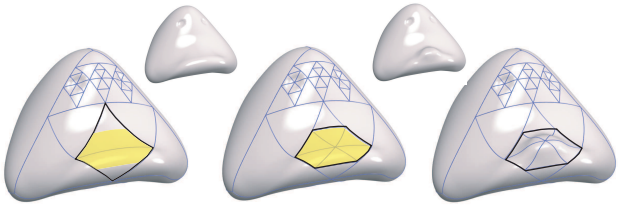


Figure 5. Local surface refinement.

- Two neighbouring patches are subdivided by four-splitting their domain triangles. The highlighted area in Figure 5(a) maps to a region that is originally composed of six triangular Bézier patches. These Bézier patches are replaced by six macro patches computed by applying locally the same triangular interpolant.
- The construction of the triangular surface ensures that the new macro patches join the remaining surface with G^1 continuity.
- The degrees of freedom which are no longer available from the original surface are the 6 inner control points of the two original macro patches.
- The newly created degrees of freedom comprise 6 inner control points for each of the six new macro patches, 6 first derivatives and 6 twists related to the newly inserted vertex of valence 6, and the position of that vertex. These new degrees of freedom are illustrated in Figure 6. The other first derivatives and twists corresponding to the 6 outer vertices of the new macro patches are fixed such as to ensure G^1 continuity with the unchanged surface around. Figure 5(c) illustrates the modification of the position of the central vertex.
- The newly inserted vertex is set to interpolate the corresponding sample point. All other new degrees of freedom are computed by minimizing (2) and (3) locally in order to decrease the error.

The adaptive fitting process therefore works as follows: Given the initial fitting surface S_0 a locally refined fitting surface S_1 is computed by refining all faces that don't meet the given error threshold. Only the newly inserted surface patches of S_1 are candidates for further refinement, since all other already meet the threshold. When the patches of a current level are done, all patches of the finer level are considered, an so on. Figure 9 shows the result of adaptive surface fitting for the Stanford bunny and Max Planck head.

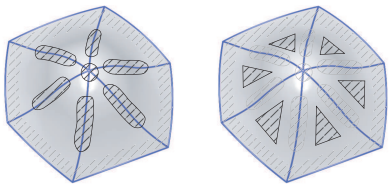


Figure 6. Additional free parameters are introduced by one surface refinement step. Left: first derivatives, twist, and position of new vertex. Right: 6 inner control points per macro patch.

4. Results

Two different models are used to illustrate the hierarchical surface fitting, the Stanford Bunny and the Max Planck Head datasets[1]. Several reconstructions with different levels of detail are computed based on these models, see Figure 7 and 8. The corresponding error measures are put in Table 2. It can be noticed that even the base surfaces with 110 and 71 faces give quite good rough approximations since the max error is less than 5%, see Fig. 7(b,c), 8(b.c). Global features are well captured at the coarse level, but the initial surface is not able to represent all the fine details. Therefore several refinements are necessary. Two successive levels are shown in Figure 7(e) and 7(h). The mean errors are divided at least by two for each new level of refinement. Details are clearly added from global to local: in Figure 7(e) bumps of the fur appear that are more precise in Figure 7(h). In Figure 7(k), the approximation error is shown. Maximum errors appear to be localized around high gradient areas of the surface. The histograms in Figure 10 and 11 underline this observation. It is therefore necessary to fit adaptively.

In Figure 9 the results of adaptive fitting are presented. The same base meshes as in Fig. 7(a) and 8(a) have been used. Starting with the initial surface (level 1) only two levels of local surface refinements are necessary to fit the models with approximately the same max error as in the case of global refinements, see Table 2. The max error is lower than 1.5% in both cases. The number of faces can be noticeably reduced, it is respectively 653 and 446 for the bunny and the Max Planck head instead of $110 \cdot 4^2 = 1760$ and $71 \cdot 4^2 = 1136$. The Max Planck head particularly benefits from this local refinement with large patches on the forehead but fine patches around high detail areas like the eyes. The ability of the present method to 4-split surface patches locally, that allows coarse surface patches to join several finer surface patches along a common boundary (see Fig-

1 Thanks to the Stanford Computer Graphics Laboratory and the Max-Planck-Institut für Informatik for providing the datasets.

Model	Bunny	Head
#faces	69473	47082
#faces of Coarse Mesh	110	71
Max Error at level 1	4.25 %	3.19 %
Max Error at level 2	2.86 %	2.11 %
Max Error at level 3	2.10 %	1.12 %
Mean Error at level 1	0.39 %	0.44 %
Mean Error at level 2	0.17 %	0.21 %
Mean Error at level 3	0.07 %	0.09 %

Table 2. Error measures for reconstructed models.

ure 9), is clearly an advantage. All surfaces are overall G^1 continuous, even along the T-junction boundary curves and vertices. It allows to refine the surface precisely only where it's needed for better fitting and thus minimizes the number of patches for a fixed error threshold.

Acknowledgements

The work was partially supported by the European Community 6-th framework programm, with the Network of Excellence *Aim@Shape* (http://www.aimatshape.net).

References

[1] M. Attene and M. Spagnuolo. Automatic surface reconstruction from point sets in space. *Computer Graphics Forum*, 19(3):457–465, 2000.

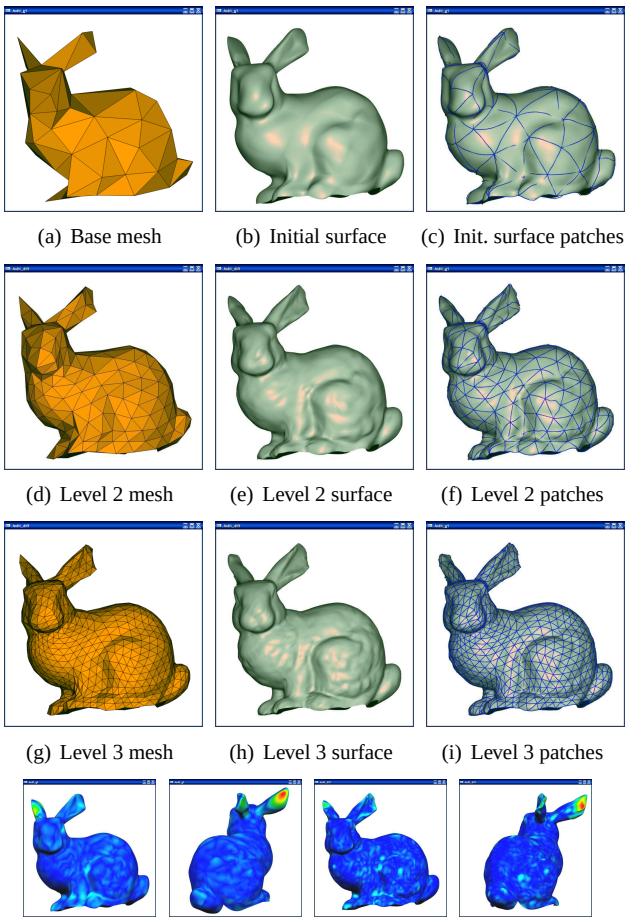
[2] C. L. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3d scans. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 109–118. ACM Press, 1995.

[3] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Trans. Graph.*, 3(4):266–286, 1984.

[4] G.-P. Bonneau and S. Hahmann. Smooth polylines on polygon meshes. In G. Brunnett, B. Hamann, H. Müller, and L. Linsen, editors, *Geometric Modeling for Scientific Visualization*, Mathematics and Visualization, pages 69–84. Springer, 2003.

[5] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM Press, 2001.

[6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. *Computer Graphics*, 30(Annual Conference Series):303–312, 1996.



(a) Base mesh (b) Initial surface (c) Init. surface patches

(d) Level 2 mesh (e) Level 2 surface (f) Level 2 patches

(g) Level 3 mesh (h) Level 3 surface (i) Level 3 patches

(j) Level 1 Error (k) from behind (l) Level 2 Error (m) from behind

Figure 7. Stanford Bunny Reconstruction.

[7] T. D. DeRose. Necessary and sufficient conditions for tangent plane continuity of bézier surfaces. *Comput. Aided Geom. Des.*, 7(1-4):165–179, 1990.

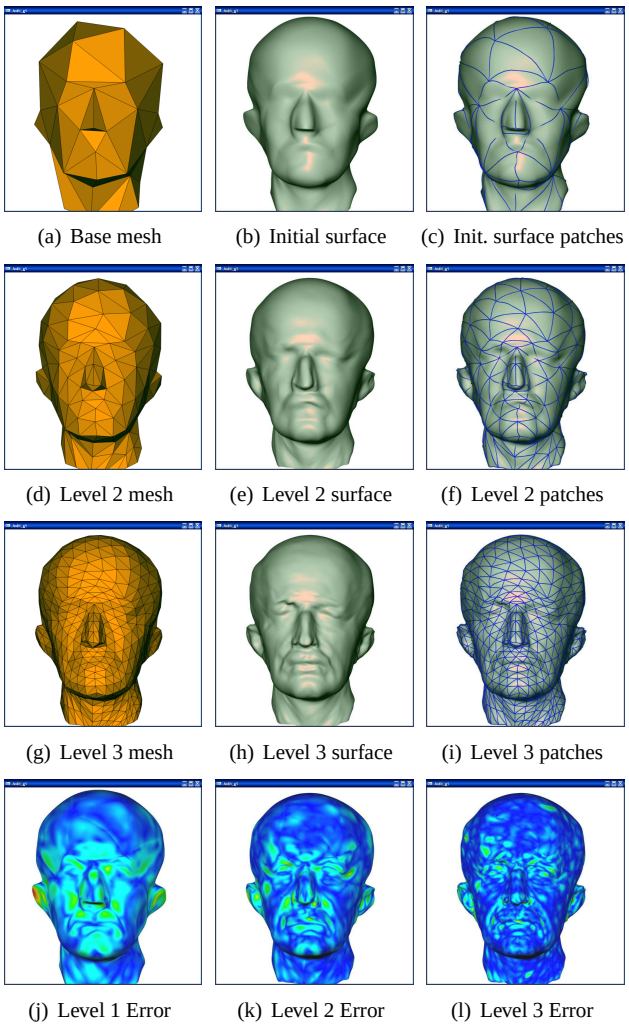
[8] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes, 2002.

[9] Y. Duan, L. Yang, H. Qin, and D. Samaras. Shape reconstruction from 3d and 2d data using pde-based deformable surfaces. In *European Conference on Computer Vision*, pages 238–251, 2004.

[10] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Computer Graphics*, 29(Annual Conference Series):173–182, 1995.

[11] M. Eck and H. Hoppe. Automatic reconstruction of b-spline surfaces of arbitrary topological type. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 325–334. ACM Press, 1996.

[12] G. Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann Publishers Inc., 2002.



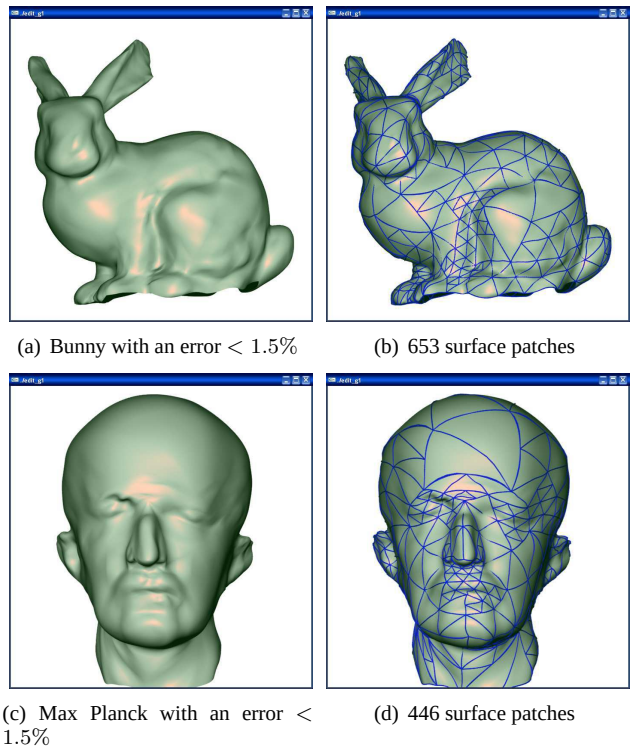
(a) Base mesh (b) Initial surface (c) Init. surface patches

(d) Level 2 mesh (e) Level 2 surface (f) Level 2 patches

(g) Level 3 mesh (h) Level 3 surface (i) Level 3 patches

(j) Level 1 Error (k) Level 2 Error (l) Level 3 Error

Figure 8. MaxPlanck Reconstruction.



(a) Bunny with an error $< 1.5\%$ (b) 653 surface patches

(c) Max Planck with an error $< 1.5\%$ (d) 446 surface patches

Figure 9. Adaptive fitting.

[13] D. R. Forsey and R. H. Bartels. Surface fitting with hierarchical splines. *ACM Transactions on Graphics*, 14(2):134–161, 1995.

[14] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. *Computer Graphics Proceedings (SIGGRAPH 97)*, 1997.

[15] S. Hahmann and G.-P. Bonneau. Polynomial surfaces interpolating arbitrary triangulations. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):99–109, 2003.

[16] M. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using catmull-clark surfaces. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 35–44. ACM Press, 1993.

[17] P. Heckbert and M. Garland. Survey on polygonal surface simplification algorithms. In *Course notes of Siggraph 97. ACM SIGGRAPH.*, 1997.

[18] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. *Computer Graphics*, 28(Annual Conference Series):295–302, 1994.

[19] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[20] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 19–26. ACM Press, 1993.

[21] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. *Computer Graphics*, 30(Annual Conference Series):313–324, 1996.

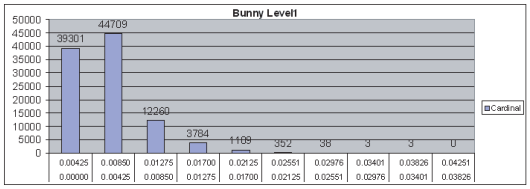
[22] A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 85–94. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[23] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. Maps: Multiresolution adaptive parameterization of surfaces. *Computer Graphics Proceedings (SIGGRAPH 98)*, pages 95–104, 1998.

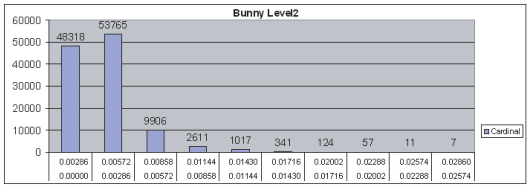
[24] C. Liao and G. Medioni. Surface approximation of a cloud of 3d points. *GMIP*, 57(1):67–74, January 1995.

[25] N. Litke, A. Levin, and P. Schroeder. Fitting subdivision surfaces. In *IEEE Visualization 2001*, pages 319–324, October 2001.

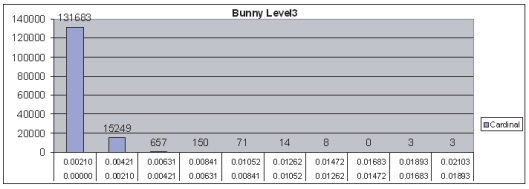
[26] M. Lounsbery, S. Mann, and T. DeRose. Parametric surface interpolation. *IEEE Comput. Graph. Appl.*, 12(5):45–52, 1992.



(a) Bunny at level 1

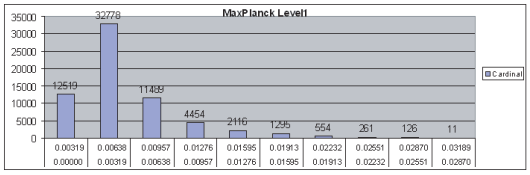


(b) Bunny at level 2

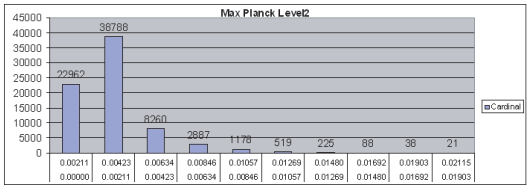


(c) Bunny at level 3

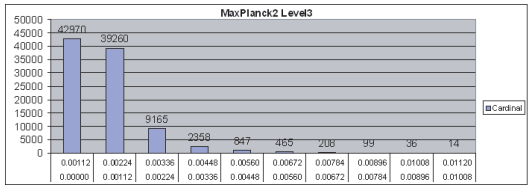
Figure 10. Error distribution for bunny model: number of points belonging to intervals of error thresholds.



(a) Max Planck at level 1



(b) Max Planck at level 2



(c) Max Planck at level 3

Figure 11. Error distribution for Max Planck model: number of points belonging to intervals of error thresholds.

[27] D. Moore and J. Warren. Approximation of dense scattered data using algebraic surfaces. In *Proc. of the 24th Annual Hawaii Intnl. Conf. on System Science, Kauai, HI, USA*, pages 681–690. IEEE Comp. Soc. Press, 1991.

[28] B. S. Morse, T. S. Yoo, D. T. Chen, P. Rheingans, and K. R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Proceedings of the International Conference on Shape Modeling & Applications*, page 89. IEEE Computer Society, 2001.

[29] J. Peters. Constructing c1 surfaces of arbitrary topology using biquadratic and bicubic splines. In *Designing Fair Curves and Surfaces*, pages 277–293. SIAM, 1994.

[30] D. Rogers and N. Fog. Constrained b-spline curve and surface fitting. *Comput. Aided Des.*, 21:641–648, 1989.

[31] B. Sarkar and C.-H. Menq. Parameter optimization in approximating curves and surfaces to measurement data. *Comput. Aided Geom. Des.*, 8(4):267–290, 1991.

[32] V. V. Savchenko, A. A. Pasko, O. G. Okunev, and T. L. Kunii. Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum*, 14(4):181–188, 1995.

[33] F. Schmitt, B. Barsky, and W. Du. An adaptive subdivision method for surface-fitting from sampled data. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 179–188. ACM Press, 1986.

[34] X. Shi, T. Wang, P. Wu, and F. Liu. Reconstruction of convergent g^1 smooth b-spline surfaces. *Computer Aided Geometric Design*, 21:893–913, 2004.

[35] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: recovering 3d shape and nongrid motion. *Artif. Intell.*, 36(1):91–123, 1988.

[36] I. Tobor, P. Reuter, and C. Schlick. Multiresolution reconstruction of implicit surfaces with attributes from large unorganized point sets. In *Proceedings of Shape Modeling International (SMI)*, pages 19–30, 2004.

[37] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318. ACM Press, 1994.

[38] R. C. Veltkamp. *Closed Object Boundaries from Scattered Points*. Springer-Verlag New York, Inc., 1994.

[39] A. Yvart, S. Hahmann, and G. Bonneau. Hierarchical triangular splines. *submitted*, 2004.