In Girod, G. Greiner, H. Niemann, H.-P. Seidel (Eds.), Proc. of Vision, Modeling and Visualization 2000, 22.-24.
November 2000, pp. 163-170, Akademische Verlagsgesellschaft Aka GmbH, Berlin

A Non-linear Subdivision Scheme for Triangle Meshes

Stefan Karbacher, Stephan Seeger, and Gerd Häusler

University of Erlangen, Chair for Optics
Staudt Str. 7/B2, D-91058 Erlangen, Germany
Email: stefan@karbacher.de, URL:http://www.karbacher.de

Abstract

Subdivision schemes are commonly used to obtain dense or smooth data representations from sparse discrete data. E. g., B-splines are smooth curves or surfaces that can be constructed by infinite subdivision of a polyline or polygon mesh of control points. New vertices are computed by *linear* combinations of the initial control points. We present a new *non-linear* subdivision scheme for the refinement of triangle meshes that generates smooth surfaces with minimum curvature variations. It is based on a combination of edge splitting operations and interpolation by blending circular arcs. In contrast to most conventional methods the final mesh density may be locally adapted to the structure of the mesh. As an application we demonstrate how this subdivision scheme can be used to reconstruct missing range data of incompletely digitized 3-D objects.

1 Introduction

The basic idea behind recursive subdivision is to create a smooth limit function by infinite refinement of an initial piecewise-linear function (see Fig. 1). Usually, this process is devided into two steps: a *splitting step*, which introduces new vertices and an *averaging* or *interpolating step*, which computes new positions for the vertices. In case of *linear* schemes, the new positions are linear combinations of the positions from the previous iteration. If the positions of the initial vertices remain unchanged, the subdivision scheme is called *interpolating*, otherwise it is *approximating*. *Uniform* subdivision applies the same splitting and averaging rule to all vertices. Some non-uniform schemes are able to *adapt* their rules to local characteristics of the piecewise-linear function. *Stationary* schemes use the same rules for all iterations. A comprehensive overview of subdivision techniques is given in [16].

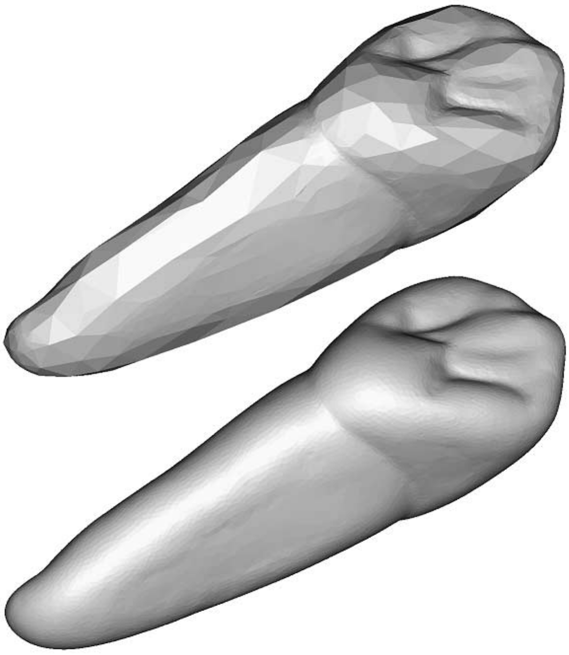

Figure 1: Subdividing a coarse triangle mesh of a human canine (upper: 9,000 triangles, lower: 28,000 triangles).

Linear subdivision can be considered as an algorithmic generalization of classical spline techniques that requires no global parametrization. For that reason, the control meshes may have arbitrary topology. The schemes are embedded into the theory of wavelets, since any collection of refinable scaling functions — a basic pre-condition for constructing wavelet spaces — can be generated by linear subdivision (see [14] for details). The limit function may be calculated directly. It is not required to perform any iterations. The first and most popular subdivision schemes for surfaces were introduced by Doo/Sabin [4] and Catmull/Clark [1]. Their methods are both based on quadrilateral meshes and either generalize biquadratic or bicubic tensor product B-splines. The simplest scheme for triangle meshes, introduced by Loop [12], splits each triangle into four and converges to quartic triangular B-Splines. All these schemes are linear, approximating, uniform and stationary. Thus, a mesh cannot be adaptively refined. However, it is always possible to apply mesh thinning techniques after subdivision (see [17] for an example).

Although it is well known that normal based geometry computations are superior to any other schemes (just compare the results of Phong and Gouraud shading), most researchers avoid such approaches, because they are non-linear and therefore difficult to handle in theory. Linear methods, however, are not well suited for modeling geometric data. For visualization and CAM techniques like CNC milling, surfaces with minimum curvature variations are usually desired. This cannot be achieved using linear techniques. Hence, our new subdivision scheme is based on a non-linear method for modeling meshes discussed in detail in [6, 8, 9, 10]. It was originally designed for dense triangle meshes like those reconstructed from optical range data, but works with sparse meshes as well. The generated surfaces feature minimum curvature variations and are therefore especially suited for visualization and rapid prototyping. In Sec. 3 we give a brief introduction to this approach.

2 Related Work

A basic ingredient of subdivision is interpolation, which requires a theoretical framework to achieve satisfying results. General triangle meshes, however, are not structured. Thus, no basic theory for interpolating, subdividing, smoothing, and compressing triangle meshes is known. Merely some limited approaches exist that work for special cases.

For regular structures (e. g. images), *multiresolution analysis* based on wavelets is well known [14]. The data are decomposed by a series of high and low pass filters. In contrast to the sine and cosine functions of Fourier analysis, the wavelet basis functions are spatially and temporally limited. Thus, finite signals are easier to process while avoiding any artifacts. The simplest type of wavelets for images are Haar functions which simply add (low pass) or subtract (high pass) neighboring pixels. Lounsbery et al. [13] have generalized this approach for semiregular meshes with subdivision connectivity: all vertices (with singular exceptions) must have the same number of neighbors. The original mesh is approximated by a coarse one that is adequate just to describe the topology of the object. Usually a few hundred triangles are sufficient. Objects which are homeomorphic to a sphere may even be approximated by a tetrahedron. A series of correction terms (wavelet coefficients) is computed. These are necessary to refine the basic mesh by recursive subdivision until the original mesh is reconstructed. Each subdivision level owns a complete record of wavelet coefficients. Structure dependent mesh reduction is simply done by eliminating small coefficients. General meshes must be remeshed in advance in order to achieve subdivision connectivity. In this case the original mesh can only be reconstructed approximately. Another drawback of this approach is the constant mesh density in each resolution step. Adaptive refinement is not possible.

A *signal processing approach* for general meshes was proposed by Taubin [15]. He generalized the discrete Fourier transform by interpreting frequencies as eigenvectors of a discrete Laplacian. Defining such a Laplacian for irregular meshes allows usage of linear signal processing tools like high and low pass filters, data compression and multiresolution hierarchies. The Laplacian can be expressed as a weighted sum of all difference vectors to all direct neighbors of a vertex (“umbrella operator,” see [11]). Laplacian smoothing and interpolation reduces surface curvature and tends to flatten the surface, so it must be used very carefully. Since the vertices are isotropically moved, the geometry

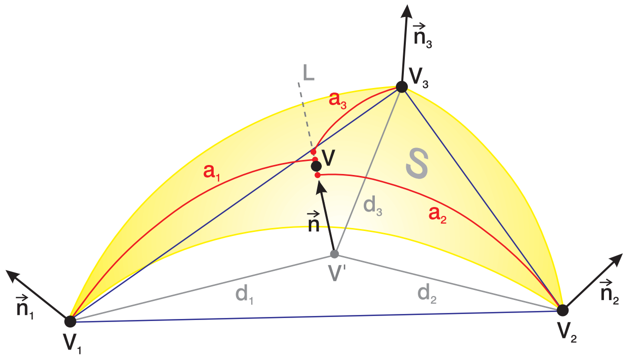

Figure 2: Interpolation of a new vertex $\mathbf{V}$ over triangle $\triangle(\mathbf{V}_1,\mathbf{V}_2,\mathbf{V}_3)$ by blending circular arcs.

may be seriously damaged, even if the surface is flat. Guskov et al. [5] introduced a more complex Laplacian that leaves flat surfaces invariant. They apply local parameterizations in order to compute second order derivatives.

In order to construct a multiresolution analysis for irregular meshes, Daubechies et al. [2] have combined the signal processing approach with Hoppe's *progressive meshes* [7]. Non-uniform "second generation wavelets" are defined for that purpose.

However, as mentioned before, the translation of concepts of linear signal theory is not the optimal choice for modeling geometry data. Therefore Desbrun et al. [3] generalized the Laplacian approach for invariance of surfaces with constant curvature. Their "curvature flow" operator for mesh smoothing is heuristically defined. Our "circular arcs" approach, in contrast, is inspired by physical observations.

3 Interpolation

Our method works best with dense meshes like those reconstructed from range images. It is assumed that the triangle mesh approximates a smooth surface with the vertices as sampled surface points. The sampling density must be high enough to neglect the variations of surface curvature between adjacent vertices. If this is true, the underlying surface can locally be approximated by circular arcs. All necessary information can be derived solely from the vertex positions and the assigned vertex normals of the triangle mesh.

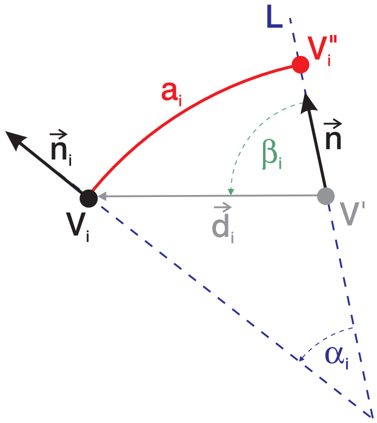

Figure 3: Cross section through one of the arcs of Fig. 2.

As an example, we show how a curved triangle is interpolated by blending circular arcs that originate from its vertices (see Fig. 2).

In order to interpolate a new vertex $\mathbf{V}$ over a flat triangle $\triangle(\mathbf{V}_1,\mathbf{V}_2,\mathbf{V}_3)$, the projection $\mathbf{V}'$ of $\mathbf{V}$ onto the triangle is constructed from given barycentric coordinates b_i:

$$\mathbf{V}' = \sum_i^3 b_i \mathbf{V}_i \tag{1}$$

with $\sum_i^3 b_i = 1$ and $b_i \geq 0$ for all i. A new surface normal $\vec{\mathbf{n}}$ is computed for that position by linear interpolation of the surrounding vertex normals $\vec{\mathbf{n}}_i$:

$$\vec{\mathbf{n}} = \frac{\sum_i^3 b_i \vec{\mathbf{n}}_i}{\|\sum_i^3 b_i \vec{\mathbf{n}}_i\|}. \tag{2}$$

The new normal $\vec{\mathbf{n}}$ defines a straight line $\mathbf{L}$ through $\mathbf{V}'$. This line and each vertex normal $\vec{\mathbf{n}}_i$ define a circular arc $\mathbf{a}_i$ with radius r_i that originates from $\mathbf{V}_i$ and intersects $\mathbf{L}$ in $\mathbf{V}_i''$:

$$r_i = \frac{\|\mathbf{a}_i\|}{\alpha_i} \approx \frac{d_i}{\alpha_i}, \tag{3}$$

$$\mathbf{V}_i'' = \mathbf{V}' + \delta_i \vec{\mathbf{n}} \tag{4}$$

with

$$\delta_i = \|\mathbf{V}_i'' - \mathbf{V}'\| \approx d_i \frac{\cos(\beta_i - \frac{\alpha_i}{2})}{\cos(\frac{\alpha_i}{2})} \tag{5}$$

(see [8] for details). As shown in Fig. 3, d_i is the distance between $\mathbf{V}$ and $\mathbf{V}_i$, α_i is the angle between the projections of the normals $\vec{\mathbf{n}}$ and $\vec{\mathbf{n}}_i$, and β_i is the angle between $\vec{\mathbf{n}}$ and the distance vector $\vec{\mathbf{d}}_i = \mathbf{V}_i - \mathbf{V}'$. The normals $\vec{\mathbf{n}}$ and $\vec{\mathbf{n}}_i$ do not need to intersect in 3-D space. Since the interpolated surface should be as constantly curved as possible, the new vertex $\mathbf{V}$ should be as close to all $\mathbf{V}_i''$ as possible. This is ensured by linear interpolation of $\mathbf{V}$ from the positions of the $\mathbf{V}_i''$:

$$\mathbf{V} = \sum_i^3 b_i \mathbf{V}_i''. \tag{6}$$

The surface $\mathbf{S}$ of the curved triangle is the infinite set of interpolated vertices $\mathbf{V}(\mathbf{b})$ for all possible parameter values $\mathbf{b} = (b_1, b_2, b_3)^T$:

$$\mathbf{V}(\mathbf{b}) = \sum_i^3 b_i \left(\mathbf{V}_i + \vec{\mathbf{n}}(\mathbf{b})\delta_i(\mathbf{b})\right). \tag{7}$$

For practical computation, δ_i can be expressed in terms of inner products $\vec{\mathbf{n}} \cdot \vec{\mathbf{n}}_i$ and $\vec{\mathbf{n}} \cdot \vec{\mathbf{d}}_i$:

$$\delta_i = d_i \cos\beta_i + \\ + d_i \sqrt{\frac{(1-\cos^2\beta_i)(1-\cos\alpha_i)}{1+\cos\alpha_i}} \tag{8}$$

with

$$\cos\alpha_i \approx \vec{\mathbf{n}} \cdot \vec{\mathbf{n}}_i, \tag{9}$$

$$\cos\beta_i \approx \frac{\vec{\mathbf{n}} \cdot \vec{\mathbf{d}}_i}{d_i}. \tag{10}$$

Unfortunately, this interpolation scheme does not generate G^1-continuous (tangent continuous) transitions between adjacent triangles, since sometimes the interpolated surface $\mathbf{S}$ does not perpendicularly intersect with the vertex normals,[1] as shown in Fig. 4. This can be enforced, however, by smoothing the final surface using the method introduced in [6, 8, 9, 10]. Since the positions of the original vertices $\mathbf{V}_i$ are moved in this case, the curved triangle *approximates* the flat one.

[1] Actually this only happens, if the curvature variations between adjacent vertices are not negligible.

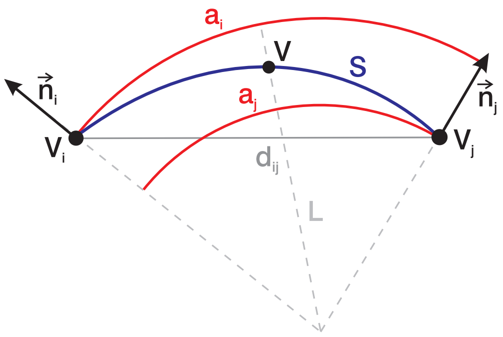

Figure 4: Cross section through one of the edges of triangle $\triangle(\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3)$. The interpolated surface $\mathbf{S}$ does not perpendicularly intersect with the vertex normals.

4 Subdivision

A given triangle is subdivided by first splitting the triangle into four new ones and then raising the new vertices with the described interpolation method (see Fig. 5).

Uniform refinement of a triangle mesh is carried out by splitting each edge of the mesh at its midpoint (see Fig. 6) and interpolating a new position for the new vertex. Thus, the splitting scheme is the same as in Loop's [12] approach. If $\mathbf{V}_1$ and $\mathbf{V}_2$ are the endpoints of the split edge, the barycentric coordinates for interpolation are $\mathbf{b} = (0.5, 0.5, 0)^T$. This procedure is iterated until a given number of iterations is reached. Figure 5 shows the first iteration of refining one single triangle and the resulting limit surface $\mathbf{S}$ for an infinite number of iterations. As in linear ap-

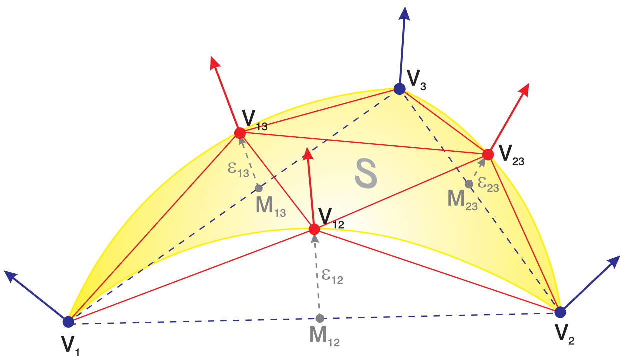

Figure 5: Splitting triangle $\triangle(\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3)$ into four new triangles by subdivision.

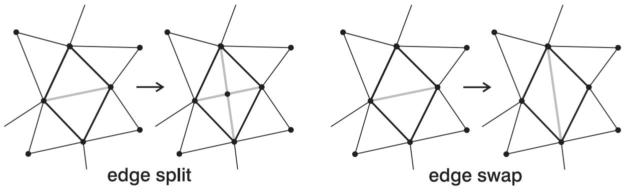

Figure 6: Topological operations for triangle meshes.

proaches, it is possible to compute the limit surface by directly applying Eq. 7 to arbitrary vertex positions **b**.

For *adaptive refinement*, the distances between the edge midpoints and the interpolated vertex positions serve as cost functions (see Fig. 5). If $\mathbf{V}_i$ and $\mathbf{V}_j$ are the endpoints of an edge with length d_{ij}, $\mathbf{M}_{ij}$ is the midpoint of this edge and $\mathbf{V}_{ij}$ the new vertex, the cost function ε_{ij} for this edge is defined by

$$\varepsilon_{ij} = \|\mathbf{V}_{ij} - \mathbf{M}_{ij}\| = \frac{1}{4} d_{ij}(\delta_i + \delta_j). \qquad (11)$$

Only those edges, for which the cost functions exceed a given threshold, are split. This threshold limits the approximation error of the final surface with regard to the limit surface.

Figures 7 and 8 display subdivision results for synthetic and noiseless data. The mesh of a cube is uniformly refined by 4 subdivision steps (15 new vertices for each initial edge), resulting in an approximation of a sphere. Due to the non-linear interpolation rule, the limit surface of a perfect (noiseless) cube is a sphere. This would not be possible using solely linear techniques. In the second example a height step is adaptively refined with an approximation error of 0.5% of the original sampling intervals. An additional constraint restricts the new sampling distances not to be less than 10% of the old values.

As mentioned above, the pure interpolation rule generates G^1-discontinuities in the limit surface (upper row of Fig. 8). These discontinuities vanish if the mesh is smoothed after each subdivision step, resulting in a mesh that *approximates* the original one (lower row of Fig. 8). In this case the limit surface **S** cannot be directly computed. It *must* be constructed by recursive subdivision.

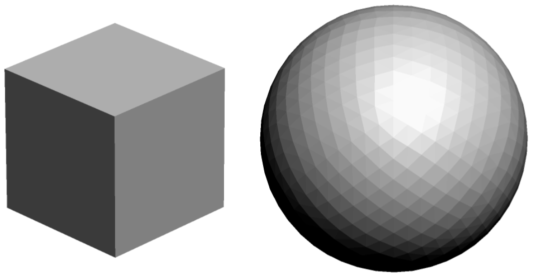

Figure 7: Uniform subdivision of a cube (4 iterations).

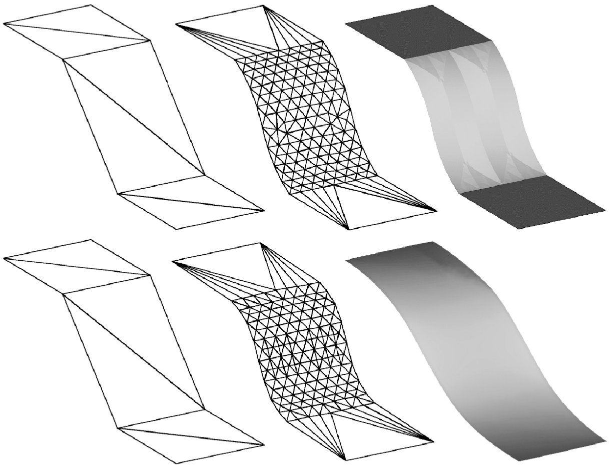

Figure 8: Adaptive subdivision of a height step (left); upper: interpolating subdivision with G^1-discontinous limit surface (right); lower: approximating subdivision with G^1-continuous limit surface (right).

While the non-linear interpolation scheme ensures the limit surface to be G^2-continuous within each initial triangle, the transitions between initial patches may only be G^1-continuous. This is not a disadvantage, because real world objects often have such discontinuities, for example when a cylindrical patch meets a flat one. A G^2-continuous transition would cause surface undulations in this case. Thus, the limit surface always adopts the continuity characteristics of the vertex normals.

Adaptive subdivision poses another problem: Since edges are irregularly split, the resulting mesh may become rather uneven with elongated or even degenerated triangles. To avoid this, a topological optimization step must be carried out

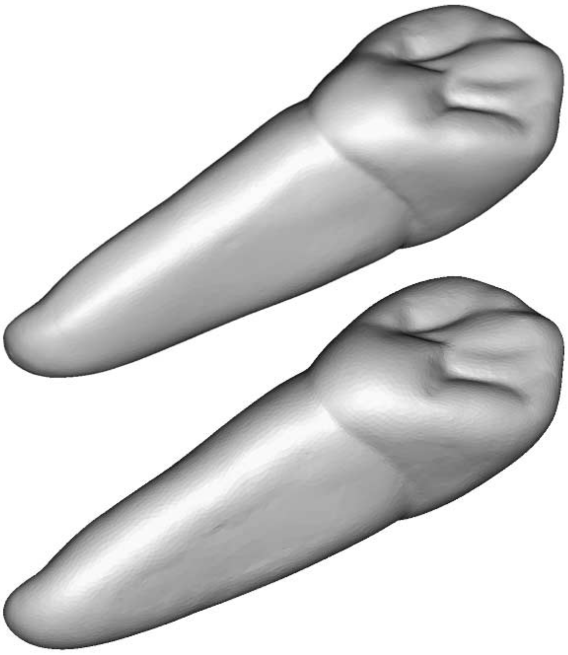

Figure 9: The original dense mesh of the human canine (upper: 41,000 triangles) and the refined one from Fig. 1 (lower: 28,000 triangles).

after each subdivision iteration. Edge swap operations (see Fig. 6) are used to keep the triangles as equilateral as possible.

The coarse mesh of Fig. 1 is a thinned version of a dense mesh that was reconstructed from 10 range images, and was refined with our approximating scheme (Fig. 9). The mean deviation between the original mesh and the refined one is 35 μm, the maximum is 315 μm. This indicates that the subdivision surface, although not perfect, is quite a good reconstruction of the original data.

5 An Application

While measuring complex objects with optical 3-D sensors, it is often difficult to scan the complete surface. Usually, data are missing due to shading effects or because it is not possible to reach all parts of the object, specifically if there are cavities and other small sized concave regions. We now demonstrate how the described subdivision method can be used to reconstruct these missing data.

Missing data cause gaps and holes in the reconstructed triangle mesh (see Fig. 10). These defects are filled with flat triangles by topological operations (see Figs. 11 and 12). Gaps between separate components of the mesh are closed by gap bridging, holes are closed by mesh growing. The mesh growth operation simply connects adjacent edges with a new one and checks if the new triangle does not overlap with

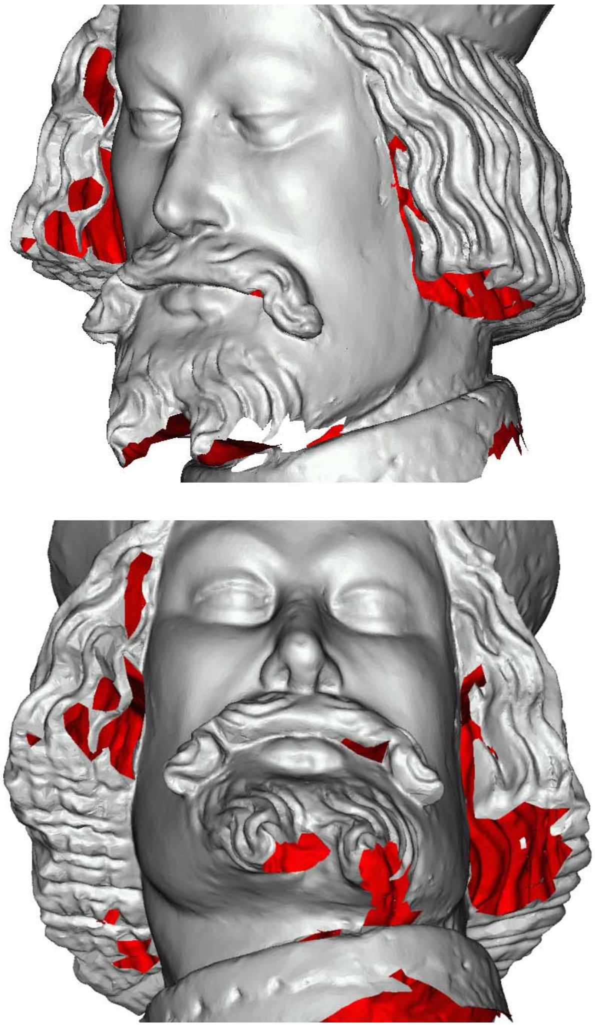

Figure 10: Mesh of St. George with holes due to missing data.

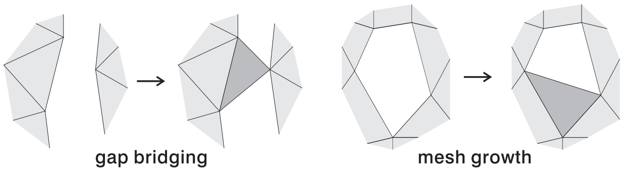

Figure 11: Topological operations to close gaps and holes in triangle meshes.

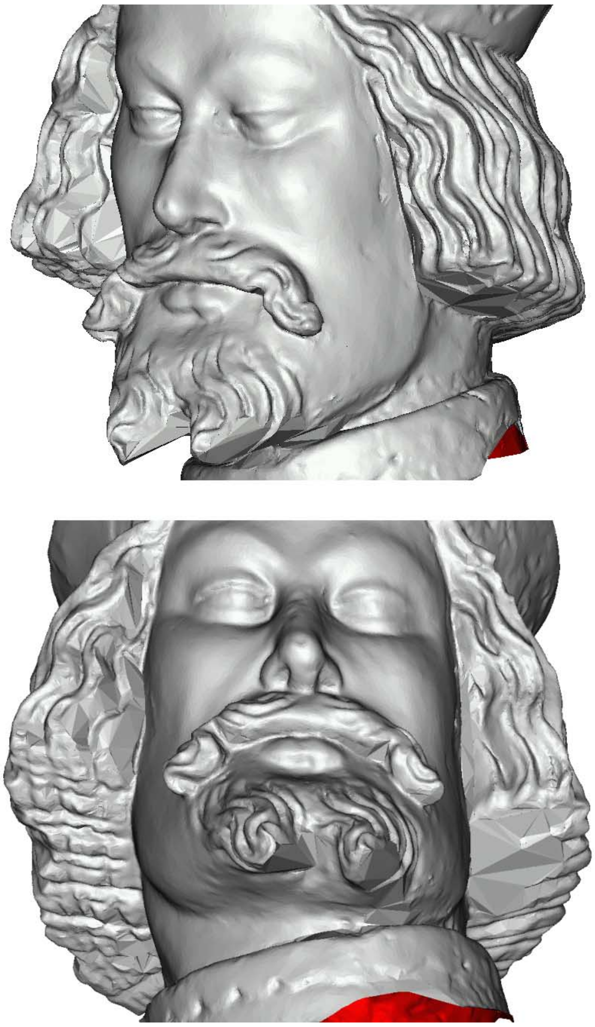

Figure 12: The holes are closed with flat triangles.

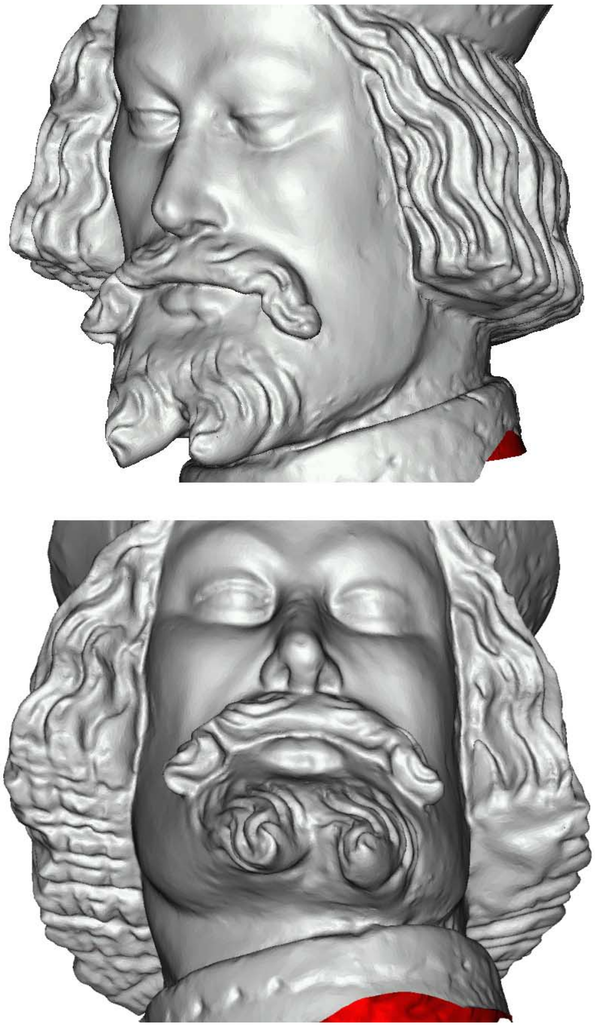

Figure 13: The new triangles are refined by subdivision.

any other triangle in its vicinity (see [8] for details). In order to bridge gaps for each edge of the first component, the closest point of the second component must be found. Normally, it is not trivial to solve this so-called “closest point problem” in less than quadratic time. However, in our case we need only search for boundary vertices. Since their number usually has the magnitude $\sqrt{N}$, even the brute force approach solves the problem in linear time. Finally, the flat triangles filled in by the closing operations are refined with the new approximating subdivision method (see Fig. 13). The holes are filled in with surfaces that extrapolate the data at the boundaries with minimum curvature variations. As a result, the transition between measured and synthetic data is not visible. The shape is reconstructed in a plausible manner. Of course, it is not possible to reconstruct small details which were not scanned.

6 Conclusions

The practical benefit of the new non-linear subdivision scheme, at least the approximating one, is obvious: It enables to seamlessly integrate synthetic surfaces into measured data. Unlike linear subdivision, which is embedded into the theoretical framework of wavelets, the theoretical background is still not satisfying. It seems to be a challenge for future research to extend the classical wavelet theory to non-linear schemes. Maybe, this will create the long desired theoretical basis for interpolation, subdivision, smoothing, and compression of unstructured data like triangle meshes.

7 Acknowledgement

This work was sponsored by "Sonderforschungsbereich 603: Model Based Analysis and Visualization of Complex Scenes and Sensor Data" of the "Deutsche Forschungsgemeinschaft" (German Research Society).

References

[1] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, September 1978.

[2] I. Daubechies, I. Guskov, P. Schröder, and W. Sweldens. Wavelets on irregular point sets. *Phil. Trans. R. Soc. Lond. A.*, 357(1760):2397–2413, 1999.

[3] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In A. Rockwood, editor, *Proceedings of SIGGRAPH 99*, pages 317–324. Addison Wesley, August 1999.

[4] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, September 1978.

[5] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. Technical Report TR-99-01, Princeton University, January 1999.

[6] G. Häusler and S. Karbacher. Reconstruction of smoothed polyhedral surfaces from multiple range images. In B. Girod, H. Niemann, and H.-P. Seidel, editors, *3D image analysis and synthesis '97*, pages 191–198. Infix Verlag, Sankt Augustin, 1997.

[7] H. Hoppe. Progressive meshes. In H. Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, pages 99–108. Addison Wesley, August 1996.

[8] S. Karbacher. *Rekonstruktion und Modellierung von Flächen aus Tiefenbildern*. Shaker Verlag, Aachen, 1997.

[9] S. Karbacher. Discrete modeling of point clouds. In B. Girod, G. Greiner, and H. Niemann, editors, *Principles of 3D Image Analysis and Synthesis*, pages 166–175. Kluwer Academic Publishers, Boston-Dordrecht-London, 2000.

[10] S. Karbacher, G. Häusler, and H. Schönfeld. Reverse engineering using optical range sensors. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook of Computer Vision and Applications*, volume 3: Systems and Applications, pages 359–380. Academic Press, Boston, 1999.

[11] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multiresolution modeling on arbitrary meshes. In M. Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, pages 105–114. Addison Wesley, July 1998.

[12] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, August 1987.

[13] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, January 1997.

[14] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. *Wavelets for Computer Graphics*. Morgan Kaufmann Publishers, Inc., 1996.

[15] G. Taubin. A signal processing approach to fair surface design. In R. Cook, editor, *SIGGRAPH 95 Conference Proceedings*, pages 351–358. Addison Wesley, August 1995.

[16] D. Zorin, P. Schröder, T. DeRose, L. Kobbelt, A. Levin, and W. Sweldens. Subdivision for modeling and animation, July 2000. SIGGRAPH 2000 Course Notes, www.mrl.nyu.edu/dzorin/sig00course/.

[17] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In T. Whitted, editor, *Proceedings of SIGGRAPH 97*, pages 259–268. Addison Wesley, August 1997.